Cryptography
Theory and Practice

Second Edition

Solution Manual

Douglas R. Stinson

Contents

0 Introduction

This is the solution manual to *Cryptography Theory and Practice, Second Edition*, which was published in March, 2002. I provide "final answers" to computational questions, and detailed proofs for all mathematical questions. I obtained most answers to computational questions using *Maple*, which is a convenient language for performing many calculations related to cryptography. Computer programs are not included in this solution manual, however.

This solution manual refers to the first printing of the book. Several exercises contained typos or other errors, which are noted in this solution manual. Also, I maintain an up-to-date errata list for the book, which can be found on the following web page:

`www.cacr.math.uwaterloo.ca/~dstinson/CTAP2/CTAP2.html`

These errors will be fixed in later printings of the book.

I would appreciate any comments or feedback about this solution manual and about the book in general, especially relating to its suitability as a textbook. In particular, I will be grateful to anyone who finds errors in the book and points them out to me.

I hope that this solution manual will be a useful resource for instructors teaching courses in cryptography. Please try to prevent the distribution of this manual to students! The usefulness of the Exercises will be severely limited if this manual somehow escapes into the public domain.

Douglas R. Stinson
Waterloo, Ontario
June, 2002

1

Classical Cryptography

Exercises

1.1 Evaluate the following:

(a) $7503 \bmod 81$.
Answer: $7503 \bmod 81 = 51$.

(b) $(-7503) \bmod 81$.
Answer: $(-7503) \bmod 81 = 45$.

(c) $81 \bmod 7503$.
Answer: $81 \bmod 7503 = 81$.

(d) $(-81) \bmod 7503$.
Answer: $(-81) \bmod 7503 = 7422$.

1.2 Suppose that $a, m > 0$, and $a \not\equiv 0 \pmod{m}$. Prove that

$$(-a) \bmod m = m - (a \bmod m).$$

Answer: We have $a = qm + r$, where $1 \leq r \leq m - 1$ and $r = a \bmod m$. Then $-a = -(q+1)m + (m - r)$, where $1 \leq m - r \leq m - 1$. Therefore $(-a) \bmod m = m - r = m - (a \bmod m)$.

1.3 Prove that $a \bmod m = b \bmod m$ if and only if $a \equiv b \pmod{m}$.
Answer: $a \bmod m = b \bmod m$ implies that $a = q_1 m + r$ and $b = q_2 m + r$, where $0 \leq r \leq m - 1$. Then $a - b = (q_1 - q_2)m$, so $a \equiv b \pmod{m}$. Conversely, suppose $a \equiv b \pmod{m}$. Then $a - b = qm$. Let $r = a \bmod m$. Then $a = q_1 m + r$ for some q_1, and hence $b = a - qm = (q_1 - q)m + r$, so $b \bmod m = r$.

1.4 Prove that $a \bmod m = a - \lfloor \frac{a}{m} \rfloor m$, where $\lfloor x \rfloor = \max\{y \in \mathbb{Z} : y \leq x\}$.
Answer: $(a - m + 1)/m \leq \lfloor \frac{a}{m} \rfloor \leq a/m$, so $a - m + 1 \leq m \lfloor \frac{a}{m} \rfloor \leq a$, and hence $0 \leq a - \lfloor \frac{a}{m} \rfloor m \leq m - 1$.

1.5 Use exhaustive key search to decrypt the following ciphertext, which was encrypted using a *Shift Cipher*:

`BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD`.

Answer: The key is 16, and the plaintext is the following:
Look, up in the air, it's a bird, it's a plane, it's Superman!

1.6 If an encryption function e_K is identical to the decryption function d_K, then the key K is said to be an *involutory key*. Find all the involutory keys in the *Shift Cipher*

over $\mathbb{Z}_{26}$.
Answer: The involutory keys are 0 and 13.

1.7 Determine the number of keys in an *Affine Cipher* over $\mathbb{Z}_m$ for $m = 30, 100$ and 1225.
Answer: $30 = 2 \times 3 \times 5$, so $\phi(30) = 1 \times 2 \times 4 = 8$. The affine cipher over $\mathbb{Z}_{30}$ has $30 \times 8 = 240$ keys.
$100 = 2^2 \times 5^2$, so $\phi(100) = (2^2 - 2)(5^2 - 5) = 40$. The affine cipher over $\mathbb{Z}_{100}$ has $100 \times 40 = 4000$ keys.
$1225 = 5^2 \times 7^2$, so $\phi(1225) = (5^2 - 5)(7^2 - 7) = 840$. The affine cipher over $\mathbb{Z}_{1225}$ has $1225 \times 840 = 1029000$ keys.

1.8 List all the invertible elements in $\mathbb{Z}_m$ for $m = 28, 33$ and 35.
Answer: The invertible elements in $\mathbb{Z}_{28}$ are 1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25 and 27.
The invertible elements in $\mathbb{Z}_{33}$ are 1, 2, 4, 5, 7, 8, 10, 13, 14, 16, 17, 19, 20, 23, 25, 26, 28, 29, 31 and 32.
The invertible elements in $\mathbb{Z}_{35}$ are 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33 and 34.

1.9 For $1 \leq a \leq 28$, determine $a^{-1} \bmod 29$ by trial and error.
Answer: $1^{-1} = 28$, $2^{-1} = 15$, $3^{-1} = 10$, $4^{-1} = 22$, $5^{-1} = 6$, $6^{-1} = 5$, $7^{-1} = 25$, $8^{-1} = 11$, $9^{-1} = 13$, $10^{-1} = 3$, $11^{-1} = 8$, $12^{-1} = 17$, $13^{-1} = 9$, $14^{-1} = 27$, $15^{-1} = 2$, $16^{-1} = 20$, $17^{-1} = 12$, $18^{-1} = 21$, $19^{-1} = 26$, $20^{-1} = 16$, $21^{-1} = 18$, $22^{-1} = 4$, $23^{-1} = 24$, $24^{-1} = 23$, $25^{-1} = 7$, $26^{-1} = 19$, $27^{-1} = 14$ and $28^{-1} = 28$.

1.10 Suppose that $K = (5, 21)$ is a key in an *Affine Cipher* over $\mathbb{Z}_{29}$.
(a) Express the decryption function $d_K(y)$ in the form $d_K(y) = a'y + b'$, where $a', b' \in \mathbb{Z}_{29}$.
Answer: $d_K(y) = 6y + 19$.
(b) Prove that $d_K(e_K(x)) = x$ for all $x \in \mathbb{Z}_{29}$.
Answer: $6(5x + 21) + 19 \equiv 30x + 145 \equiv x \pmod{29}$.

1.11 (a) Suppose that $K = (a, b)$ is a key in an *Affine Cipher* over $\mathbb{Z}_n$. Prove that K is an involutory key if and only if $a^{-1} \bmod n = a$ and $b(a + 1) \equiv 0 \pmod{n}$.
Answer: $K = (a, b)$ is an involutory key if and only if $a(ax + b) + b \equiv x \pmod{n}$ for all $x \in \mathbb{Z}_n$. Clearly $a(ax+b)+b \equiv a^2x+b(a+1) \pmod{n}$, so we require that $a^2 \equiv 1 \pmod{n}$ and $b(a + 1) \equiv 0 \pmod{n}$.
(b) Determine all the involutory keys in the *Affine Cipher* over $\mathbb{Z}_{15}$.
Answer: $a^2 \equiv 1 \pmod{15}$ if and only if $a = 1, 4, 11$ or 14. If $a = 1$, then $b = 0$. If $a = 4$, then $b = 0, 3, 6, 9$ or 12. If $a = 11$, then $b = 0, 5$ or 10. Finally, if $a = 14$, then b can be any element of $\mathbb{Z}_{15}$.
(c) Suppose that $n = pq$, where p and q are distinct odd primes. Prove that the number of involutory keys in the *Affine Cipher* over $\mathbb{Z}_n$ is $n + p + q + 1$.
Answer: There are four possible values for a, namely, $a = 1$; $a = -1 \bmod n$; the solution to the system $a \equiv 1 \pmod{p}$, $a \equiv -1 \pmod{q}$; and the solution to the system $a \equiv -1 \pmod{p}$, $a \equiv 1 \pmod{q}$. If $a = 1$, then $b = 0$. If $a = -1$, then b can be any element in $\mathbb{Z}_n$. In the third case, we require that $b \equiv 0 \pmod{q}$, so there are p possible values for b. In the fourth case, we require that $b \equiv 0 \pmod{p}$, so there are q possible values for b. The total number of involutory keys is therefore $n + p + q + 1$.

1.12 (a) Let p be prime. Prove that the number of 2×2 matrices that are invertible over $\mathbb{Z}_p$ is $(p^2-1)(p^2-p)$.

HINT Since p is prime, $\mathbb{Z}_p$ is a field. Use the fact that a matrix over a field is invertible if and only if its rows are linearly independent vectors (i.e., there does not exist a non-zero linear combination of the rows whose sum is the vector of all 0's).

Answer: The first row can be any non-zero vector, so there are p^2-1 possiblilities. Given the first row, say r, the second row can be any vector that is not a scalar multiple of r. Therefore there are p^2-p possibilities for the second row, given the first row. Hence, the total number of 2×2 invertible matrices is $(p^2-1)(p^2-p)$.

(b) For p prime and $m \geq 2$ an integer, find a formula for the number of $m \times m$ matrices that are invertible over $\mathbb{Z}_p$.
Answer: The number of invertible matrices is

$$(p^m-1)(p^m-p)(p^m-p^2)\cdots(p^m-p^{m-1}).$$

1.13 For $n = 6, 9$ and 26, how many 2×2 matrices are there that are invertible over $\mathbb{Z}_n$?
Answer: For $n = 6$, there are $(2^2-1)(2^2-2)(3^2-1)(3^2-3) = 1728$ invertible matrices (use the Chinese remainder theorem and Exercise 1.12). Similarly, for $n = 26$, there are $(2^2-1)(2^2-2)(13^2-1)(13^2-13) = 157248$ invertible matrices. For $n = 9$, there are $3^4(3^2-1)(3^2-3) = 3888$ invertible matrices.

1.14 (a) Prove that $\det A \equiv \pm 1 \pmod{26}$ if A is a matrix over $\mathbb{Z}_{26}$ such that $A = A^{-1}$.
Answer: If $A = A^{-1}$, then $A^2 = I$ and hence $(\det A)^2 \equiv 1 \pmod{26}$. This implies that $\det A \equiv \pm 1 \pmod{26}$.

(b) Use the formula given in Corollary 1.4 to determine the number of involutory keys in the *Hill Cipher* (over $\mathbb{Z}_{26}$) in the case $m = 2$.
Answer: If $\det A \equiv 1 \pmod{26}$ then there are 8 involutory matrices, and if $\det A \equiv -1 \pmod{26}$ then there are 728 involutory matrices, for a total of 736 involutory matrices.

The eight involutory matrices with determinant 1 are as follows:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 25 & 0 \\ 0 & 25 \end{pmatrix}, \quad \begin{pmatrix} 1 & 13 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 25 & 13 \\ 0 & 25 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 \\ 13 & 1 \end{pmatrix}, \quad \begin{pmatrix} 25 & 0 \\ 13 & 25 \end{pmatrix}, \quad \begin{pmatrix} 12 & 13 \\ 13 & 12 \end{pmatrix}, \quad \begin{pmatrix} 14 & 13 \\ 13 & 14 \end{pmatrix}.$$

The involutory matrices with determinant -1 have the following forms when reduced modulo 2:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

When reduced modulo 13, an involutory matrix with determinant -1 has the following form:

$$\begin{pmatrix} x & y \\ z & -x \end{pmatrix},$$

where $x^2 + yz \equiv 1 \pmod{13}$. The number of triples $(x, y, z) \in (\mathbb{Z}_{13})^3$ that satisfy this congruence is easily computed: if $x = 2$ or 12, then there are 25 ordered pairs (y, z); and if $x \neq 2, 12$, then there are 12 ordered pairs

(y, z). Hence, the total number of triples is $2 \times 25 + 11 \times 12 = 182$. Now we can use the Chinese remainder theorem to combine any solution modulo 2 with any solution modulo 13, so the total number of solutions modulo 26 is $4 \times 182 = 728$, as stated above.

1.15 Determine the inverses of the following matrices over $\mathbb{Z}_{26}$:

(a) $\begin{pmatrix} 11 & 15 \\ 1 & 20 \end{pmatrix}$

Answer: The inverse matrix is

$$\begin{pmatrix} 2 & 5 \\ 9 & 5 \end{pmatrix}.$$

(b) $\begin{pmatrix} 1 & 11 & 12 \\ 4 & 23 & 2 \\ 17 & 15 & 9 \end{pmatrix}$

Answer: The inverse matrix is

$$\begin{pmatrix} 25 & 11 & 22 \\ 10 & 13 & 4 \\ 17 & 24 & 1 \end{pmatrix}.$$

1.16 (a) Suppose that π is the following permutation of $\{1, \ldots, 8\}$:

x	1	2	3	4	5	6	7	8
$\pi(x)$	4	1	6	2	7	3	8	5

Compute the permutation π^{-1}.

Answer: The permutation π^{-1} is as follows:

x	1	2	3	4	5	6	7	8
$\pi^{-1}(x)$	2	4	6	1	8	3	5	7

(b) Decrypt the following ciphertext, for a *Permutation Cipher* with $m = 8$, which was encrypted using the key π:

`ETEGENLMDNTNEOORDAHATECOESAHLRMI.`

Answer: Note: This ciphertext was actually encrypted using the key π^{-1}.

The plaintext is the following:

Gentlemen do not read each other's mail.

1.17 (a) Prove that a permutation π in the *Permutation Cipher* is an involutory key if and only if $\pi(i) = j$ implies $\pi(j) = i$, for all $i, j \in \{1, \ldots, m\}$.

Answer: A permutation π is involutory if and only if $\pi(\pi(i)) = i$ for all i. Denoting $\pi(i) = j$, it must be the case that $\pi(j) = i$.

(b) Determine the number of involutory keys in the *Permutation Cipher* for $m = 2, 3, 4, 5$ and 6.

Answer: An involutory permutation must consist of fixed points and cycles of length two.

For $m = 2$, there are 2 involutory permutations.

For $m = 3$, there are 4 involutory permutations.

For $m = 4$, there are 3 permutations consisting of two cycles of length 2; 6 permutations having one cycle of length 2 and two fixed points; and 1 permutation consisting of 4 fixed points. The total number of involutory permutations is 10.

For $m = 5$, there are 15 permutations consisting of two cycles of length 2 and one fixed point; 10 permutations having one cycle of length 2 and three fixed points; and 1 permutation consisting of 5 fixed points. The total number of involutory permutations is 26.

For $m = 6$, there are 15 permutations consisting of three cycles of length 2; 45 permutations consisting of two cycles of length 2 and two fixed points; 15 permutations having one cycle of length 2 and four fixed points; and 1 permutation consisting of 5 fixed points. The total number of involutory permutations is 76.

1.18 Consider the following linear recurrence over $\mathbb{Z}_2$ of degree four:

$$z_{i+4} = (z_i + z_{i+1} + z_{i+2} + z_{i+3}) \bmod 2,$$

$i \geq 0$. For each of the 16 possible initialization vectors $(z_0, z_1, z_2, z_4) \in (\mathbb{Z}_2)^4$, determine the period of the resulting keystream.

Answer: $(0, 0, 0, 0)$ produces a keystream with period 1, and all other initialization vectors produce a keystream with period 5.

1.19 Redo the preceding question, using the recurrence

$$z_{i+4} = (z_i + z_{i+3}) \bmod 2,$$

$i \geq 0$.

Answer: $(0, 0, 0, 0)$ produces a keystream with period 1, and all other initialization vectors produce a keystream with period 15.

1.20 Suppose we construct a keystream in a synchronous stream cipher using the following method. Let $K \in \mathcal{K}$ be the key, let $\mathcal{L}$ be the keystream alphabet, and let Σ be a finite set of *states*. First, an *initial state* $\sigma_0 \in \Sigma$ is determined from K by some method. For all $i \geq 1$, the state σ_i is computed from the previous state σ_{i-1} according to the following rule:

$$\sigma_i = f(\sigma_{i-1}, K),$$

where $f : \Sigma \times \mathcal{K} \rightarrow \Sigma$. Also, for all $i \geq 1$, the keystream element z_i is computed using the following rule:

$$z_i = g(\sigma_i, K),$$

where $g : \Sigma \times \mathcal{K} \rightarrow \mathcal{L}$. Prove that any keystream produced by this method has period at most $|\Sigma|$.

Answer: For a fixed key K, each σ_i can be regarded as a function of σ_{i-1}. Define

$$t = \min\{i \geq 1 : \sigma_i \in \{\sigma_0, \ldots, \sigma_{i-1}\}\}.$$

It follows from the pigeon-hole principle that $t \leq |\Sigma|$, because $\sigma_i \in \Sigma$ for all $i \geq 0$. Suppose that $\sigma_t = \sigma_s$, where $0 \leq s < t$. Then it $\sigma_{i+t-s} = \sigma_i$ for all $i \geq s$. Hence, $z_{i+t-s} = z_i$ for all $i \geq s$, and the keystream has period $t - s \leq |\Sigma|$.

1.21 Below are given four examples of ciphertext, one obtained from a *Substitution Cipher*, one from a *Vigenère Cipher*, one from an *Affine Cipher*, and one unspecified. In each case, the task is to determine the plaintext.

Give a clearly written description of the steps you followed to decrypt each ciphertext. This should include all statistical analysis and computations you performed.

The first two plaintexts were taken from “The Diary of Samuel Marchbanks,” by Robertson Davies, Clarke Irwin, 1947; the fourth was taken from “Lake Wobegon Days,” by Garrison Keillor, Viking Penguin, Inc., 1985.

(a) *Substitution Cipher*:

```
EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCK
QPKUGKMGOLICGINCGACKSNISACYKZSCKXECJCKSHYSXCG
OIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZU
GFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNS
ACIGOIYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCFZCCNC
IACZEJNCSHFZEJZEGMXCYHCJUMGKUCY
```

HINT F decrypts to w.

Answer: The plaintext is as follows:

> I may not be able to grow flowers, but my garden produces just as many dead leaves, old overshoes, pieces of rope, and bushels of dead grass as anybody's, and today I bought a wheelbarrow to help in clearing it up. I have always loved and respected the wheelbarrow. It is the one wheeled vehicle of which I am perfect master.

(b) *Vigenère Cipher*:

```
KCCPKBGUFDPHQTYAVINRRTMVGRKDNBVFDETDGILTXRGUD
DKOTFMBPVGEGLTGCKQRACQCWDNAWCRXIZAKFTLEWRPTYC
QKYVXCHKFTPONCQQRHJVAJUWETMCMSPKQDYHJVDAHCTRL
SVSKCGCZQQDZXGSFRLSWCWSJTBHAFSIASPRJAHKJRJUMV
GKMITZHFPDISPZLVLGWTFPLKKEBDPGCEBSHCTJRWXBAFS
PEZQNRWXCVYCGAONWDDKACKAWBBIKFTIOVKCGGHJVLNHI
FFSQESVYCLACNVRWBBIREPBBVFEXOSCDYGZWPFDTKFQIY
CWHJVLNHIQIBTKHJVNPIST
```

Answer: The keyword is $CRYPTO$, and the plaintext is as follows:

> I learned how to calculate the amount of paper needed for a room when I was at school. You multiply the square footage of the walls by the cubic contents of the floor and ceiling combined, and double it. You then allow half the total for openings such as windows and doors. Then you allow the other half for matching the pattern. Then you double the whole thing again to give a margin of error, and then you order the paper.

(c) *Affine Cipher*:

```
KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI
```

Answer: The key is $(19,4)$. The plaintext consists of the French lyrics to "O Canada":

> Ô Canada!
> Terre de nos aïeux.
> Ton front est ceint,
> De fleurons glorieux.
> Car ton bras
> Sait porter l'épée,
> Il sait porter la croix.
> Ton histoire est une épopée,

des plus brillants exploits.
Et ta valeur,
de foi trempée,
protègera nos foyers et nos droits.

(d) unspecified cipher:

```
BNVSNSIHQCEELSSKKYERIFJKXUMBGYKAMQLJTYAVFBKVT
DVBPVVRJYYLAOKYMPQSCGDLFSRLLPROYGESEBUUALRWXM
MASAZLGLEDFJBZAVVPXWICGJXASCBYEHOSNMULKCEAHTQ
OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC
GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJMBLR
FFJELHWEYLWISTFVVYFJCMHYUYRUFSFMGESIGRLWALSWM
NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAUM
ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHFRAYISYSGKVSUU
HYHGGCKTMBLRX
```

Answer: This is a *Vigenère Cipher*. The keyword is $THEORY$, and the plaintext is as follows:

> I grew up among slow talkers, men in particular, who dropped words a few at a time like beans in a hill, and when I got to Minneapolis where people took a Lake Wobegon comma to mean the end of a story, I couldn't speak a whole sentence in company and was considered not too bright. So I enrolled in a speech course taught by Orville Sand, the founder of reflexive relaxology, a self-hypnotic technique that enabled a person to speak up to three hundred words per minute.

1.22 (a) Suppose that $p_1, \dots, p_n$ and $q_1, \dots, q_n$ are both probability distributions, and $p_1 \geq \cdots \geq p_n$. Let $q'_1, \dots, q'_n$ be any permutation of $q_1, \dots, q_n$. Prove that the quantity

$$\sum_{i=1}^{n} p_i q'_i$$

is maximized when $q'_1 \geq \cdots \geq q'_n$.

Answer: Suppose that $q'_j < q'_k$ for some $j < k$. Define

$$q''_i = \begin{cases} q'_i & \text{if } i \not\in \{j,k\} \\ q'_k & \text{if } i = j \\ q'_j & \text{if } i = k. \end{cases}$$

Then we have

$$\sum_{i=1}^{n} p_i q''_i - \sum_{i=1}^{n} p_i q'_i = (p_j - p_k)(q'_k - q'_j) \geq 0.$$

Therefore the desired sum is not decreased when q'_j and q'_k are exchanged. By a sequence of exchanges of this type, we see that the sum attains its maximum possible value when $q'_1 \geq \cdots \geq q'_n$.

(b) Explain why the expression in Equation (1.1) is likely to be maximized when $g = k_i$.

Answer: (Note: this equation is on page 34.) Suppose that π is a permutation of $\{0, \dots, 25\}$ such that $p_{\pi(0)} \geq \cdots \geq p_{\pi(25)}$. Then it is "likely" that $f_{\pi(0)} \geq \cdots \geq f_{\pi(25)}$. Assuming that this is the case, we proceed. When

$g = 0$, the following equation holds:

$$\sum_{i=0}^{25} \frac{p_i f_i}{n'} = \sum_{i=0}^{25} \frac{p_{\pi(i)} f_{\pi(i)}}{n'}.$$

By the result proven in part (a), this sum is at least as great as any sum

$$\sum_{i=0}^{25} \frac{p_i f_{i+g}}{n'},$$

where $g \neq 0$.

1.23 Suppose we are told that the plaintext

`breathtaking`

yields the ciphertext

`UPOTENTOIFV`

where the *Hill Cipher* is used (but m is not specified). Determine the encryption matrix.

Answer: There is an error in the statement of this question; the plaintext does not have the same length as the ciphertext. The ciphertext should be as follows:

`RUPOTENTOIFV`

Then, using the first 9 plaintext and ciphertext characters, we compute

$$K = \begin{pmatrix} 1 & 17 & 4 \\ 0 & 19 & 7 \\ 19 & 0 & 10 \end{pmatrix}^{-1} \begin{pmatrix} 17 & 20 & 15 \\ 14 & 19 & 4 \\ 13 & 19 & 14 \end{pmatrix} = \begin{pmatrix} 3 & 21 & 20 \\ 4 & 15 & 23 \\ 6 & 14 & 5 \end{pmatrix}.$$

If desired, we can check this by verifying that the last 3 plaintext characters encrypt properly:

$$\begin{pmatrix} 8 & 13 & 6 \end{pmatrix} \begin{pmatrix} 3 & 21 & 20 \\ 4 & 15 & 23 \\ 6 & 14 & 5 \end{pmatrix} = \begin{pmatrix} 8 & 5 & 21 \end{pmatrix}.$$

1.24 An *Affine-Hill Cipher* is the following modification of a *Hill Cipher*: Let m be a positive integer, and define $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$. In this cryptosystem, a key K consists of a pair (L, b), where L is an $m \times m$ invertible matrix over $\mathbb{Z}_{26}$, and $b \in (\mathbb{Z}_{26})^m$. For $x = (x_1, \ldots, x_m) \in \mathcal{P}$ and $K = (L, b) \in \mathcal{K}$, we compute $y = e_K(x) = (y_1, \ldots, y_m)$ by means of the formula $y = xL + b$. Hence, if $L = (\ell_{i,j})$ and $b = (b_1, \ldots, b_m)$, then

$$(y_1, \ldots, y_m) = (x_1, \ldots, x_m) \begin{pmatrix} \ell_{1,1} & \ell_{1,2} & \ldots & \ell_{1,m} \\ \ell_{2,1} & \ell_{2,2} & \ldots & \ell_{2,m} \\ \vdots & \vdots & & \vdots \\ \ell_{m,1} & \ell_{m,2} & \ldots & \ell_{m,m} \end{pmatrix} + (b_1, \ldots, b_m).$$

Suppose Oscar has learned that the plaintext

`adisplayedequation`

is encrypted to give the ciphertext

`DSRMSIOPLXLJBZULLM`

and Oscar also knows that $m = 3$. Determine the key, showing all computations.
Answer: We are given the following:

$$\begin{aligned} x_1 &= (0,3,8)\\ x_2 &= (18,15,11)\\ x_3 &= (0,24,4)\\ x_4 &= (3,4,16)\\ x_5 &= (20,0,9)\\ x_6 &= (8,14,13)\end{aligned}$$

and

$$\begin{aligned} y_1 &= (3,18,17)\\ y_2 &= (12,18,8)\\ y_3 &= (14,15,11)\\ y_4 &= (23,11,9)\\ y_5 &= (1,25,20)\\ y_6 &= (11,11,12).\end{aligned}$$

For $1 \leq i \leq 6$, it holds that $y_i = x_i L + b$. Therefore, for $1 \leq i \leq 3$, we have $y_i - y_4 = (x_i - x_4)L$. We form the 3×3 matrix X' having rows $x_i - x_4$ ($1 \leq i \leq 3$) and the 3×3 matrix Y' having rows $y_i - y_4$ ($1 \leq i \leq 3$); then $L = (X')^{-1}Y'$. Once we have found L, we can determine b from the equation $b = y_1 - x_1 L$.

In the given example, we have

$$X' = \begin{pmatrix} 23 & 25 & 18\\ 15 & 11 & 21\\ 23 & 20 & 14\end{pmatrix},$$

$$Y' = \begin{pmatrix} 6 & 7 & 8\\ 15 & 7 & 25\\ 17 & 4 & 2\end{pmatrix},$$

and L can be computed to be

$$L = \begin{pmatrix} 3 & 6 & 4\\ 5 & 15 & 18\\ 17 & 8 & 5\end{pmatrix}.$$

Then

$$b = \begin{pmatrix} 8 & 13 & 1\end{pmatrix}.$$

If desired, it can be checked that $y_i = x_i L + b$, for $1 \leq i \leq 6$.

1.25 Here is how we might cryptanalyze the *Hill Cipher* using a ciphertext-only attack. Suppose that we know that $m = 2$. Break the ciphertext into blocks of length two letters (digrams). Each such digram is the encryption of a plaintext digram using the unknown encryption matrix. Pick out the most frequent ciphertext digram and assume it is the encryption of a common digram in the list following Table 1.1 (for example, TH or ST). For each such guess, proceed as in the known-plaintext attack, until the correct encryption matrix is found.

Here is a sample of ciphertext for you to decrypt using this method:

```
LMQETXYEAGTXCTUIEWNCTXLZEWUAISPZYVAPEWLMGQWYA
XFTCJMSQCADAGTXLMDXNXSNPJQSYVAPRIQSMHNOCVAXFV
```

Answer: The key is

$$\begin{pmatrix} 4 & 11 \\ 13 & 9 \end{pmatrix}.$$

The plaintext is the following:

> The king was in his counting house, counting out his money. The queen was in the parlour, eating bread and honey.

1.26 We describe a special case of a *Permutation Cipher.* Let m, n be positive integers. Write out the plaintext, by rows, in $m \times n$ rectangles. Then form the ciphertext by taking the columns of these rectangles. For example, if $m = 4, n = 3$, then we would encrypt the plaintext "$cryptography$" by forming the following rectangle:

```
cryp
togr
aphy
```

The ciphertext would be "$CTAROPYGHPRY$."

(a) Describe how Bob would decrypt a ciphertext string (given values for m and n).

Answer: Bob can write out the ciphertext string by rows, in $n \times m$ rectangles. The plaintext is formed by taking the columns of these rectangles.

(b) Decrypt the following ciphertext, which was obtained by using this method of encryption:

```
MYAMRARUYIQTENCTORAHROYWDSOYEOUARRGDERNOGW
```

Answer: Here $m = 2$ and $n = 3$. The plaintext is the following:

> Mary, Mary, quite contrary, how does your garden grow?

1.27 The purpose of this exercise is to prove the statement made in Section 1.2.5 that the $m \times m$ coefficient matrix is invertible. This is equivalent to saying that the rows of this matrix are linearly independent vectors over $\mathbb{Z}_2$.

As before, we suppose that the recurrence has the form

$$z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2.$$

$(z_1, \dots, z_m)$ comprises the initialization vector. For $i \geq 1$, define

$$v_i = (z_i, \dots, z_{i+m-1}).$$

Note that the coefficient matrix has the vectors $v_1, \dots, v_m$ as its rows, so our objective is to prove that these m vectors are linearly independent.

Prove the following assertions:

(a) For any $i \geq 1$,

$$v_{m+i} = \sum_{j=0}^{m-1} c_j v_{i+j} \bmod 2.$$

Answer: This is immediate.

(b) Choose h to be the minimum integer such that there exists a non-trivial linear combination of the vectors $v_1, \dots, v_h$ which sums to the vector $(0, \dots, 0)$ modulo 2. Then

$$v_h = \sum_{j=0}^{h-2} \alpha_j v_{j+1} \bmod 2,$$

and not all the α_j's are zero. Observe that $h \leq m+1$, since any $m+1$ vectors in an m-dimensional vector space are dependent.
Answer: A dependence relation has the form

$$\sum_{j=0}^{h-1} \alpha_j v_{j+1} \bmod 2 = (0, \ldots, 0),$$

where $\alpha_0, \ldots, \alpha_{h-1} \in \{0,1\}$. Clearly $h \leq m+1$, because any $m+1$ vectors are linearly dependent. Also, we note that $\alpha_{h-1} = 1$ by the minimality of h. Therefore

$$v_h = \sum_{j=0}^{h-2} \alpha_j v_{j+1} \bmod 2.$$

Now, could it be the case that $\alpha_0 = \cdots = \alpha_{h-2} = 0$? If so, then we have $v_h = (0, \ldots, 0)$. But $v_h = (z_h, \ldots, z_{h+m-1})$, so $z_h = \cdots = z_{h+m-1} = 0$. Using the fact that $c_0 = 1$ (as discussed in Section 1.1.7), we can rewrite the recurrence

$$z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

"backwards", as follows:

$$z_i = \sum_{j=1}^{m} c_j z_{i+j} \bmod 2,$$

where we define $c_m = 1$. Then we see that $z_1 = \cdots = z_m = 0$, which generates a keystream consisting entirely of "0"s. We do not allow this case to occur (as discussed on page 22), which proves the desired result.

(c) Prove that the keystream must satisfy the recurrence

$$z_{h-1+i} = \sum_{j=0}^{h-2} \alpha_j z_{j+i} \bmod 2$$

for any $i \geq 1$.
Answer: This is immediate.

(d) Observe that if $h \leq m$, then the keystream satisfies a linear recurrence of degree less than m, a contradiction. Hence, $h = m+1$, and the matrix must be invertible.
Answer: In part (c), we showed that the keystream satisfies a recurrence of degree at most $h-1$. However, the keystream is generated by a recurrence of degree exactly equal to m, which implies that it cannot be generated by a recurrence of lower degree. Hence $h-1 = m$. Therefore the m vectors $v_1, \ldots, v_m$ are linearly independent, and the matrix is invertible.

1.28 Decrypt the following ciphertext, obtained from the *Autokey Cipher*, by using exhaustive key search:

MALVVMAFBHBUQPTSOXALTGVWWRG

Answer: The key is 19, and the plaintext is the following:
There is no time like the present.

1.29 We describe a stream cipher that is a modification of the *Vigenère Cipher*. Given a keyword $(K_1, \ldots, K_m)$ of length m, construct a keystream by the rule $z_i = K_i$ ($1 \leq i \leq m$), $z_{i+m} = (z_i + 1) \bmod 26$ ($i \geq 1$). In other words, each time we use the keyword, we replace each letter by its successor modulo 26. For example, if $SUMMER$ is the keyword, we use $SUMMER$ to encrypt the first six letters, we use $TVNNFS$ for the next six letters, and so on.

(a) Describe how you can use the concept of index of coincidence to first determine the length of the keyword, and then actually find the keyword.
Answer: Suppose we hypothesize that the keyword length is m. Define the following modified ciphertext:

$$y'_j = y_j - \left\lfloor \frac{j-1}{m} \right\rfloor ,$$

$j = 1, 2, \dots$. Then the string $y'_1 y'_2 \cdots$ is the encryption of the same plaintext, using the usual *Vigenère Cipher* with the same keyword. Hence the methods used to cryptanalyze the *Vigenère Cipher* can be applied to this modified ciphertext string to determine the keyword length and the actual keyword.

(b) Test your method by cryptanalyzing the following ciphertext:

```
IYMYSILONRFNCQXQJEDSHBUIBCJUZBOLFQYSCHATPEQGQ
JEJNGNXZWHHGWFSUKULJQACZKKJOAAHGKEMTAFGMKVRDO
PXNEHEKZNKFSKIFRQVHHOVXINPHMRTJPYWQGJWPUUVKFP
OAWPMRKKQZWLQDYAZDRMLPBJKJOBWIWPSEPVVQMBCRYVC
RUZAAOUMBCHDAGDIEMSZFZHALIGKEMJJFPCIWKRMLMPIN
AYOFIREAOLDTHITDVRMSE
```

Answer: Tke keyword is $PRIME$. The plaintext is from page 351 of ”The Codebreakers”, by D. Kahn, Macmillan, 1967.

> The most famous cryptologist in history owes his fame less to what he did than to what he said, and to the sensational way in which he said it, and this was most perfectly in character, for Herbert Osborne Yardley was perhaps the most engaging, articulate, and technicolored personality in the business.

1.30 We describe another stream cipher, which incorporates one of the ideas from the “Enigma” system used by Germany in World War II. Suppose that π is a fixed permutation of $\mathbb{Z}_{26}$. The key is an element $K \in \mathbb{Z}_{26}$. For all integers $i \geq 1$, the keystream element $z_i \in \mathbb{Z}_{26}$ is defined according to the rule $z_i = (K + i - 1) \bmod 26$. Encryption and decryption are performed using the permutations π and π^{-1}, respectively, as follows:

$$e_z(x) = \pi((x + z) \bmod 26)$$

and

$$d_z(y) = (\pi^{-1}(y) - z) \bmod 26,$$

where $z \in \mathbb{Z}_{26}$.

Suppose that π is the following permutation of $\mathbb{Z}_{26}$:

x	0	1	2	3	4	5	6	7	8	9	10	11	12
$\pi(x)$	23	13	24	0	7	15	14	6	25	16	22	1	19

x	13	14	15	16	17	18	19	20	21	22	23	24	25
$\pi(x)$	18	5	11	17	2	21	12	20	4	10	9	3	8

The following ciphertext has been encrypted using this stream cipher; use exhaustive key search to decrypt it:

```
WRTCNRLDSAFARWKXFTXCZRNHNYPDTZUUKMPLUSOXNEUDO
KLXRMCBKGRCCURR
```

Answer: The encryption and decryption rules are written incorrectly. They should be as follows:

$$e_z(x) = \pi(x) + z \bmod 26$$

and

$$d_z(y) = \pi^{-1}(y - z \bmod 26),$$

The key is $K = 10$, and the decrypted plaintext is the following:

> The first deposit consisted of one thousand and fourteen pounds of gold.

2

Shannon's Theory

Exercises

2.1 Referring to Example 2.2, determine all the joint and conditional probabilities, $\mathbf{Pr}[x, y]$, $\mathbf{Pr}[x|y]$ and $\mathbf{Pr}[y|x]$, where $x \in \{2, \ldots, 12\}$ and $y \in \{D, N\}$.
Answer: The probabilities are as follows:

x	y	$\mathbf{Pr}[x\|y]$	$\mathbf{Pr}[y\|x]$	$\mathbf{Pr}[x, y]$
2	D	$1/6$	1	$1/36$
3	D	0	0	0
4	D	$1/6$	$1/3$	$1/36$
5	D	0	0	0
6	D	$1/6$	$1/5$	$1/36$
7	D	0	0	0
8	D	$1/6$	$1/5$	$1/36$
9	D	0	0	0
10	D	$1/6$	$1/3$	$1/36$
11	D	0	0	0
12	D	$1/6$	1	$1/36$
2	N	0	0	0
3	N	$2/30$	1	$2/36$
4	N	$2/30$	$2/3$	$2/36$
5	N	$4/30$	1	$4/36$
6	N	$4/30$	$4/5$	$4/36$
7	N	$6/30$	1	$6/36$
8	N	$4/30$	$4/5$	$4/36$
9	N	$4/30$	1	$4/36$
10	N	$2/30$	$2/3$	$2/36$
11	N	$2/30$	1	$2/36$
12	N	0	0	0

2.2 Let n be a positive integer. A *Latin square* of order n is an $n \times n$ array L of the integers $1, \ldots, n$ such that every one of the n integers occurs exactly once in each row and each column of L. An example of a Latin square of order 3 is as follows:

1	2	3
3	1	2
2	3	1

Given any Latin square L of order n, we can define a related cryptosystem. Take $\mathcal{P} = \mathcal{C} = \mathcal{K} = \{1, \ldots, n\}$. For $1 \leq i \leq n$, the encryption rule e_i is defined to be $e_i(j) = L(i, j)$. (Hence each row of L gives rise to one encryption rule.)

Give a complete proof that this *Latin Square Cryptosystem* achieves perfect secrecy provided that every key is used with equal probability.

Answer: For each $x, y \in \{1, \ldots, n\}$, there exists a unique key $K_{x,y}$ such that $e_{K_{x,y}}(x) = y$. Therefore, $C(K) = \{1, \ldots, n\}$ for all $K \in \mathcal{K}$. For any $y \in \{1, \ldots, n\}$, we have

$$\begin{aligned}\mathbf{Pr}[\mathbf{y} = y] &= \sum_{x \in \{1,\ldots,n\}} \mathbf{Pr}[\mathbf{K} = K_{x,y}]\mathbf{Pr}[\mathbf{x} = x] \\ &= \sum_{x \in \{1,\ldots,n\}} (1/n) \times \mathbf{Pr}[\mathbf{x} = x] \\ &= \frac{1}{n}.\end{aligned}$$

Then, for any $x, y \in \{1, \ldots, n\}$, we compute

$$\mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x] = \mathbf{Pr}[\mathbf{K} = K_{x,y}] = \frac{1}{n}.$$

Finally, using Bayes' Theorem, we see that

$$\mathbf{Pr}[\mathbf{x} = x | \mathbf{y} = y] = \mathbf{Pr}[\mathbf{x} = x]$$

for all x, y.

2.3 (a) Prove that the *Affine Cipher* achieves perfect secrecy if every key is used with equal probability $1/312$.

Answer: For each $x, y \in \mathbb{Z}_{26}$, and for each $a \in {\mathbb{Z}_{26}}^*$, there exists a unique $b(x, y, a) \in \mathbb{Z}_{26}$ such that $e_{(a, b(x,y,a))}(x) = y$. Also, $C(K) = \{1, \ldots, n\}$ for all $K \in \mathcal{K}$. For any $y \in \{1, \ldots, n\}$, we have

$$\begin{aligned}\mathbf{Pr}[\mathbf{y} = y] &= \sum_{x \in \{1,\ldots,n\}} \sum_{a \in {\mathbb{Z}_{26}}^*} \mathbf{Pr}[\mathbf{K} = (a, b(x, y, a))]\mathbf{Pr}[\mathbf{x} = x] \\ &= \sum_{x \in \{1,\ldots,n\}} (12/312) \times \mathbf{Pr}[\mathbf{x} = x] \\ &= \frac{1}{26}.\end{aligned}$$

Then, for any $x, y \in \mathbb{Z}_{26}$, we compute

$$\begin{aligned}\mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x] &= \sum_{a \in {\mathbb{Z}_{26}}^*} \mathbf{Pr}[\mathbf{K} = (a, b(x, y, a))] \\ &= \frac{12}{312} \\ &= \frac{1}{26}.\end{aligned}$$

Finally, using Bayes' Theorem, we see that

$$\mathbf{Pr}[\mathbf{x} = x | \mathbf{y} = y] = \mathbf{Pr}[\mathbf{x} = x]$$

for all x, y.

(b) More generally, suppose we are given a probability distribution on the set

$$\{a \in \mathbb{Z}_{26} : \gcd(a, 26) = 1\}.$$

Suppose that every key (a, b) for the *Affine Cipher* is used with probability $1/(26 \times \mathbf{Pr}[a])$. Prove that the *Affine Cipher* achieves perfect secrecy when this probability distribution is defined on the keyspace.

Answer: The question is stated incorrectly: The probability of key (a, b) should be $\mathbf{Pr}[a]/26$.

Proceeding as in part (a), for any $y \in \{1, \dots, n\}$, we have

$$\begin{aligned}
\mathbf{Pr}[\mathbf{y} = y] &= \sum_{x \in \{1,\dots,n\}} \sum_{a \in \mathbb{Z}_{26}{}^*} \mathbf{Pr}[\mathbf{K} = (a, b(x, y, a))]\mathbf{Pr}[\mathbf{x} = x] \\
&= \sum_{x \in \{1,\dots,n\}} \sum_{a \in \mathbb{Z}_{26}{}^*} (\mathbf{Pr}[a]/26) \times \mathbf{Pr}[\mathbf{x} = x] \\
&= \sum_{x \in \{1,\dots,n\}} (1/26) \times \mathbf{Pr}[\mathbf{x} = x] \\
&= \frac{1}{26}.
\end{aligned}$$

Then, for any $x, y \in \mathbb{Z}_{26}$, we compute

$$\begin{aligned}
\mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x] &= \sum_{a \in \mathbb{Z}_{26}{}^*} \mathbf{Pr}[\mathbf{K} = (a, b(x, y, a))] \\
&= \sum_{a \in \mathbb{Z}_{26}{}^*} \frac{\mathbf{Pr}[a]}{26} \\
&= \frac{1}{26}.
\end{aligned}$$

Finally, using Bayes' Theorem, we see that

$$\mathbf{Pr}[\mathbf{x} = x | \mathbf{y} = y] = \mathbf{Pr}[\mathbf{x} = x]$$

for all x, y.

2.4 Suppose a cryptosystem achieves perfect secrecy for a particular plaintext probability distribution. Prove that perfect secrecy is maintained for any plaintext probability distribution.

Answer: Let $\mathsf{p} = p_{x_1}, \dots, p_{x_n}$ be a probability distribution on the plaintext space $\mathcal{P} = \{x_1, \dots, x_n\}$, and suppose that the cryptosystem achieves perfect secrecy when the plaintext is chosen using this plaintext probability distribution. Let $\mathsf{q} = q_{x_1}, \dots, q_{x_n}$ be an arbitrary probability distribution on $\mathcal{P}$. It should be clear that $\mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x]$ does not depend on the plaintext probability distribution.

Because the perfect secrecy property holds with respect to p, we have that

$$\mathbf{Pr}_{\mathsf{p}}[\mathbf{y} = y] = \mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x]$$

for all $x \in \mathcal{P}$, $y \in \mathcal{Y}$. Therefore it holds that

$$\sum_{\{K : y \in C(K)\}} (\mathbf{Pr}[\mathbf{K} = K] \times p_{d_K(y)}) = \sum_{\{K : x = d_K(y)\}} \mathbf{Pr}[\mathbf{K} = K]$$

for all $x \in \mathcal{P}$, $y \in \mathcal{Y}$. Now, we compute $\mathbf{Pr}_{\mathsf{q}}[\mathbf{y} = y]$:

$$\begin{aligned}
\mathbf{Pr}_{\mathsf{q}}[\mathbf{y} = y] &= \sum_{\{K : y \in C(K)\}} (\mathbf{Pr}[\mathbf{K} = K] \times q_{d_K(y)}) \\
&= \sum_{x_i \in \mathcal{P}} q_{x_i} \sum_{\{K : x_i = d_K(y)\}} \mathbf{Pr}[\mathbf{K} = K] \\
&= \sum_{x_i \in \mathcal{P}} q_{x_i} \sum_{\{K : y \in C(K)\}} (\mathbf{Pr}[\mathbf{K} = K] \times p_{d_K(y)}) \\
&= \left(\sum_{x_i \in \mathcal{P}} q_{x_i} \right) \times \left(\sum_{\{K : y \in C(K)\}} (\mathbf{Pr}[\mathbf{K} = K] \times p_{d_K(y)}) \right) \\
&= \sum_{\{K : y \in C(K)\}} (\mathbf{Pr}[\mathbf{K} = K] \times p_{d_K(y)}) \\
&= \mathbf{Pr}[\mathbf{y} = y | \mathbf{x} = x],
\end{aligned}$$

as desired.

2.5 Prove that if a cryptosystem has perfect secrecy and $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$, then every ciphertext is equally probable.

Answer: This follows from the proof of Theorem 2.4.

2.6 Suppose that y and y' are two ciphertext elements (i.e., binary n-tuples) in the *One-time Pad* that were obtained by encrypting plaintext elements x and x', respectively, using the same key, K. Prove that $x + x' \equiv y + y' \pmod 2$.

Answer: We have $y = x + K \bmod 2$ and $y' = x' + K \bmod 2$. Adding, we see that

$$y + y' = x + K + x' + K = x + x' \bmod 2.$$

2.7 (a) Construct the encryption matrix (as defined in Example 2.3) for the *One-time Pad* with $n = 3$.

Answer:

K	$x =$ 000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	001	000	011	010	101	100	111	110
010	010	011	000	001	110	111	100	101
011	011	010	001	000	111	110	101	100
100	100	101	110	111	000	001	010	011
101	101	100	111	110	001	000	011	010
110	110	111	100	101	010	011	000	001
111	111	110	101	100	011	010	001	000

(b) For any positive integer n, give a direct proof that the encryption matrix of a *One-time Pad* defined over $(\mathbb{Z}_2)^n$ is a Latin square of order n.

Answer: This is a misprint. The encryption matrix is a Latin square of order 2^n, in which the symbols are the elements of the group $(\mathbb{Z}_2)^n$.

Suppose that $x, y, K \in (\mathbb{Z}_2)^n$. We have that $e_K(x) = y$ if and only if $x + K = y$ (in $(\mathbb{Z}_2)^n$). Given K and y, we can solve for x uniquely: $x = y + K$. Therefore every row of the encryption matrix contains every symbol in exactly one cell. Given x and y, we can solve for K uniquely:

$K = x + y$. Therefore every column of the encryption matrix contains every symbol in exactly one cell.

2.8 Suppose X is a set of cardinality n, where $2^k \leq n < 2^{k+1}$, and $\mathbf{Pr}[x] = 1/n$ for all $x \in X$.

(a) Find a prefix-free encoding of X, say f, such that $\ell(f) = k + 2 - 2^{k+1}/n$.

HINT Encode $2^{k+1} - n$ elements of X as strings of length k, and encode the remaining elements as strings of length $k + 1$.

Answer: Let Y be the set of all 2^k binary strings of length k. Let $Z_1 \subseteq Y$, $|Z_1| = 2^{k+1} - n$. Then, for each string $y \in Y \backslash Z_1$, construct two strings, $y \parallel 0$ and $y \parallel 1$, and call the resulting set of $2(2^k - (2^{k+1} - n)) = 2n - 2^{k+1}$ strings Z_2. Then the set $Z = Z_1 \cup Z_2$ is a set of n strings that satisfies the prefix-free property, so it is a Huffman Code. We can define a Huffman encoding of X by taking f to be any bijection from X to Z.

It is now straightforward to compute $\ell(f)$:

$$\begin{aligned}\ell(f) &= \frac{1}{n}((2^{k+1} - n)k + (2n - 2^{k+1})(k+1)) \\ &= \frac{1}{n}(n(k+2) - 2^{k+1}) \\ &= k + 2 - \frac{2^{k+1}}{n}.\end{aligned}$$

(b) Illustrate your construction for $n = 6$. Compute $\ell(f)$ and $H(\mathbf{X})$ in this case.
Answer: Here we have $n = 6$ and $k = 2$. The binary strings of length 2 are 00, 01, 10 and 11. Suppose we take $Z_1 = \{00, 01\}$. Then we form $Z_2 = \{100, 101, 110, 111\}$, and $Z = \{00, 01, 100, 101, 110, 111\}$. Here we have $\ell(f) = 8/3 \approx 2.67$ and $H(\mathbf{X}) = \log_2 6 \approx 2.58$.

2.9 Suppose $X = \{a, b, c, d, e\}$ has the following probability distribution: $\mathbf{Pr}[a] = .32$, $\mathbf{Pr}[b] = .23$, $\mathbf{Pr}[c] = .20$, $\mathbf{Pr}[d] = .15$ and $\mathbf{Pr}[e] = .10$. Use Huffman's algorithm to find the optimal prefix-free encoding of X. Compare the length of this encoding to $H(\mathbf{X})$.
Answer: We obtain the following Huffman encoding:

x	$f(x)$
a	100
b	101
c	00
d	01
e	11

Thus, the average length encoding is

$$\begin{aligned}\ell(f) &= .1 \times 3 + .15 \times 3 + .20 \times 2 + .23 \times 2 + .32 \times 2 \\ &= 2.25.\end{aligned}$$

The entropy is

$$\begin{aligned}H(\mathbf{X}) &= .3322 + .4105 + .4644 + .4877 + .5260 \\ &= 2.2208.\end{aligned}$$

2.10 Prove that $H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{Y}) + H(\mathbf{X}|\mathbf{Y})$. Then show as a corollary that $H(\mathbf{X}|\mathbf{Y}) \leq H(\mathbf{X})$, with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent.

Answer: First, we observe that

$$\mathbf{Pr}[y]\mathbf{Pr}[x|y] = \mathbf{Pr}[x,y]$$

and

$$\log_2(\mathbf{Pr}[x|y]) = \log_2(\mathbf{Pr}[x,y]/\mathbf{Pr}[y]) = \log_2 \mathbf{Pr}[x,y] - \log_2 \mathbf{Pr}[y].$$

Therefore

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) &= -\sum_y \sum_x \mathbf{Pr}[y]\mathbf{Pr}[x|y] \log_2 \mathbf{Pr}[x|y] \\ &= -\sum_y \sum_x \mathbf{Pr}[x,y](\log_2 \mathbf{Pr}[x,y] - \log_2 \mathbf{Pr}[y]) \\ &= H(\mathbf{X},\mathbf{Y}) + \sum_y \sum_x \mathbf{Pr}[x,y] \log_2 \mathbf{Pr}[y] \\ &= H(\mathbf{X},\mathbf{Y}) + \sum_y \log_2 \mathbf{Pr}[y] \left(\sum_x \mathbf{Pr}[x,y] \right) \\ &= H(\mathbf{X},\mathbf{Y}) + \sum_y (\log_2 \mathbf{Pr}[y] \times \mathbf{Pr}[y]) \\ &= H(\mathbf{X},\mathbf{Y}) - H(\mathbf{Y}), \end{aligned}$$

as desired.

Theorem 2.7 says that $H(\mathbf{X},\mathbf{Y}) \leq H(\mathbf{X}) + H(\mathbf{Y})$, with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. Therefore we have

$$\begin{aligned} H(\mathbf{X}) + H(\mathbf{Y}) &\geq H(\mathbf{X},\mathbf{Y}) \\ &= H(\mathbf{Y}) + H(\mathbf{X}|\mathbf{Y}), \end{aligned}$$

which implies that $H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y})$. Further equality occurs if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent.

2.11 Prove that a cryptosystem has perfect secrecy if and only if $H(\mathbf{P}|\mathbf{C}) = H(\mathbf{P})$.
Answer: From Exercise 2.9, we have that $H(\mathbf{P}|\mathbf{C}) = H(\mathbf{P})$ if and only if $\mathbf{P}$ and $\mathbf{C}$ are independent. This is true if and only if $\mathbf{Pr}[x,y] = \mathbf{Pr}[x]\mathbf{Pr}[y]$ for all $x \in \mathcal{P}$ and all $y \in \mathcal{C}$. Writing $\mathbf{Pr}[x,y] = \mathbf{Pr}[x|y]\mathbf{Pr}[y]$, the condition becomes $\mathbf{Pr}[x|y]\mathbf{Pr}[y] = \mathbf{Pr}[x]\mathbf{Pr}[y]$, which simplifies to $\mathbf{Pr}[x|y] = \mathbf{Pr}[x]$. This is precisely the perfect secrecy condition.

2.12 Prove that, in any cryptosystem, $H(\mathbf{K}|\mathbf{C}) \geq H(\mathbf{P}|\mathbf{C})$. (Intuitively, this result says that, given a ciphertext, the opponent's uncertainty about the key is at least as great as his uncertainty about the plaintext.)
Answer: Theorem 2.10 says that

$$H(\mathbf{K}|\mathbf{C}) = H(\mathbf{K}) + H(\mathbf{P}) - H(\mathbf{C}).$$

Then we compute a bound on $H(\mathbf{P}|\mathbf{C})$ as follows:

$$\begin{aligned} H(\mathbf{P}|\mathbf{C}) &= H(\mathbf{P},\mathbf{C}) - H(\mathbf{C}) \\ &= H(\mathbf{K},\mathbf{P},\mathbf{C}) - H(\mathbf{K}|\mathbf{P},\mathbf{C}) - H(\mathbf{C}) \\ &\leq H(\mathbf{K},\mathbf{P},\mathbf{C}) - H(\mathbf{C}) \\ &= H(\mathbf{K},\mathbf{P}) - H(\mathbf{C}) \\ &\leq H(\mathbf{K}) + H(\mathbf{P}) - H(\mathbf{C}) \\ &= H(\mathbf{K}|\mathbf{C}). \end{aligned}$$

2.13 Consider a cryptosystem in which $\mathcal{P} = \{a, b, c\}$, $\mathcal{K} = \{K_1, K_2, K_3\}$ and $\mathcal{C} = \{1, 2, 3, 4\}$. Suppose the encryption matrix is as follows:

	a	b	c
K_1	1	2	3
K_2	2	3	4
K_3	3	4	1

Given that keys are chosen equiprobably, and the plaintext probability distribution is $\mathbf{Pr}[a] = 1/2$, $\mathbf{Pr}[b] = 1/3$, $\mathbf{Pr}[c] = 1/6$, compute $H(\mathbf{P})$, $H(\mathbf{C})$, $H(\mathbf{K})$, $H(\mathbf{K}|\mathbf{C})$ and $H(\mathbf{P}|\mathbf{C})$.

Answer: From the given probability distributions on $\mathcal{K}$ and $\mathcal{P}$, we have $H(\mathbf{K}) = 1.585$ and $H(\mathbf{P}) = 1.459$. We next compute the probability distribution on $\mathcal{C}$ to be $\mathbf{Pr}[1] = 4/18$, $\mathbf{Pr}[2] = 5/18$, $\mathbf{Pr}[3] = 6/18$ and $\mathbf{Pr}[4] = 3/18$. Then $H(\mathbf{C}) = 1.955$. Next, we compute

$$H(\mathbf{K}|\mathbf{C}) = H(\mathbf{K}) + H(\mathbf{P}) - H(\mathbf{C}) = 1.089.$$

In order to compute $H(\mathbf{P}|\mathbf{C})$, we first compute $\mathbf{Pr}[x|y]$ for all $x \in \mathcal{P}$ and all $y \in \mathcal{C}$:

	a	b	c
1	$3/4$	0	$1/4$
2	$3/5$	$2/5$	0
3	$1/2$	$1/3$	$1/6$
4	0	$2/3$	$1/3$

From this, we compute $H(\mathbf{P}|1) = .8113$, $H(\mathbf{P}|2) = .9710$, $H(\mathbf{P}|3) = 1.459$ and $H(\mathbf{P}|4) = .9183$. Finally,

$$H(\mathbf{P}|\mathbf{C}) = \left(\frac{4}{18}, \frac{5}{18}, \frac{6}{18}, \frac{3}{18}\right) \cdot (.8113, .9710, 1.459, 9183) = 1.062.$$

2.14 Compute $H(\mathbf{K}|\mathbf{C})$ and $H(\mathbf{K}|\mathbf{P},\mathbf{C})$ for the *Affine Cipher*.

Answer: Note: here, you should assume that keys are used equiprobably, and that the plaintext probability distribution is equiprobable. Then $H(\mathbf{K}|\mathbf{C}) = \log_2 312$ and $H(\mathbf{K}|\mathbf{P},\mathbf{C}) = \log_2 12$.

2.15 Consider a *Vigenère Cipher* with keyword length m. Show that the unicity distance is $1/R_L$, where R_L is the redundancy of the underlying language. (This result is interpreted as follows. If n_0 denotes the number of alphabetic characters being encrypted, then the "length" of the plaintext is n_0/m, since each plaintext element consists of m alphabetic characters. So, a unicity distance of $1/R_L$ corresponds to a plaintext consisting of m/R_L alphabetic characters.)

Answer: In the *Vigenère Cipher*, we have $|\mathcal{P}| = |\mathcal{K}| = 26^m$, so the estimate for the unicity distance is

$$\frac{\log_2 |\mathcal{K}|}{R_L \log_2 |\mathcal{P}|} = \frac{1}{R_L}.$$

2.16 Show that the unicity distance of the *Hill Cipher* (with an $m \times m$ encryption matrix) is less than m/R_L. (Note that the number of alphabetic characters in a plaintext of this length is m^2/R_L.)

Answer: The number of $m \times m$ matrices with entries from $\mathbb{Z}_{26}$ is 26^{m^2}, but not all of these matrices are invertible. Therefore $|\mathcal{K}| < 26^{m^2}$. Also, $|\mathcal{P}| = 26^m$. The estimate for the unicity distance is

$$\frac{\log_2 |\mathcal{K}|}{R_L \log_2 |\mathcal{P}|} < \frac{m^2(\log_2 26)}{m(\log_2 26) R_L} = \frac{m}{R_L}.$$

2.17 A *Substitution Cipher* over a plaintext space of size n has $|\mathcal{K}| = n!$ Stirling's formula gives the following estimate for $n!$:

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

(a) Using Stirling's formula, derive an estimate of the unicity distance of the *Substitution Cipher.*

Answer: We have that

$$\log_2 |\mathcal{K}| \approx \log_2 \left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right) = n(\log_2 n - c_1) + 0.5 \log_2 n + c_2,$$

where c_1, c_2 are small positive constants. $\log_2 |\mathcal{P}| = \log_s n$, so an estimate for the unicity distance is

$$\frac{1}{r_L}\left(n - \frac{c_1 n}{\log_2 n} + \frac{1}{2}\right) < \frac{n}{r_L}.$$

(b) Let $m \geq 1$ be an integer. The m*-gram Substitution Cipher* is the *Substitution Cipher* where the plaintext (and ciphertext) spaces consist of all 26^m m-grams. Estimate the unicity distance of the m*-gram Substitution Cipher* if $R_L = 0.75$.

Answer: To simplify things, we will use the estimate $\log_2(n!) \approx n \log_2 n$. Setting $n = 26^m$, we get

$$\log_2 |\mathcal{K}| \approx 26^m \log_2(26^m) \approx (\log_2 26) m 26^m.$$

$\log_2 |\mathcal{P}| = (\log_2 26)m$, so the estimate for the unicity distance is

$$\frac{(\log_2 26) m 26^m}{r_L (\log_2 26) m} \approx 1.33 \times 26^m.$$

2.18 Prove that the *Shift Cipher* is idempotent.

Answer: Note: in this question, you should assume that keys are chosen equiprobably.

A key in the *Shift Cipher* is an element $K \in \mathbb{Z}_{26}$, and the corresponding encryption rule is $e_K(x) = x + K \bmod 26$ for all $x \in \mathbb{Z}_{26}$. It is clear that $e_{K_2}(e_{K_1}(x))) = x + K_1 + K_2 \bmod 26$, so the composition of two encryption rules, with keys K_1 and K_2, is another encryption rule in the *Shift Cipher*, namely the one with key $K_1 + K_2 \bmod 26$.

We need to show that the probability of each key K in the product cipher is $1/26$.

This is shown as follows:

$$\begin{aligned}
&\mathbf{Pr}[\mathbf{K_1} + \mathbf{K_2} \bmod 26 = K] \\
&= \sum_{\{(K_1,K_2)\in\mathbb{Z}_{26}\times\mathbb{Z}_{26}: K\equiv K_1+K_2 \pmod{26}\}} (\mathbf{Pr}[\mathbf{K_1} = K_1] \times \mathbf{Pr}[\mathbf{K_2} = K_2]) \\
&= \sum_{K_2\in\mathbb{Z}_{26}} (\mathbf{Pr}[\mathbf{K_1} = K - K_2 \bmod 26] \times \mathbf{Pr}[\mathbf{K_1} = K_2]) \\
&= 26 \times \frac{1}{26^2} \\
&= \frac{1}{26},
\end{aligned}$$

as desired.

2.19 Suppose $\mathbf{S}_1$ is the *Shift Cipher* (with equiprobable keys, as usual) and $\mathbf{S}_2$ is the *Shift Cipher* where keys are chosen with respect to some probability distribution $p_{\mathcal{K}}$ (which need not be equiprobable). Prove that $\mathbf{S}_1 \times \mathbf{S}_2 = \mathbf{S}_1$.

Answer: In this question, the probability computation in the previous exercise should be modified, as follows:

$$\begin{aligned}
&\mathbf{Pr}[\mathbf{K_1} + \mathbf{K_2} \bmod 26 = K] \\
&= \sum_{\{(K_1,K_2)\in\mathbb{Z}_{26}\times\mathbb{Z}_{26}: K\equiv K_1+K_2 \pmod{26}\}} (\mathbf{Pr}[\mathbf{K_1} = K_1] \times \mathbf{Pr}[\mathbf{K_2} = K_2]) \\
&= \sum_{K_2\in\mathbb{Z}_{26}} (\mathbf{Pr}[\mathbf{K_1} = K - K_2 \bmod 26] \times \mathbf{Pr}[\mathbf{K_2} = K_2]) \\
&= \sum_{K_2\in\mathbb{Z}_{26}} \frac{1}{26} \times \mathbf{Pr}[\mathbf{K_2} = K_2]) \\
&= \frac{1}{26}.
\end{aligned}$$

2.20 Suppose $\mathbf{S}_1$ and $\mathbf{S}_2$ are *Vigenère Ciphers* with keyword lengths m_1, m_2 respectively, where $m_1 > m_2$.

(a) If $m_2 \mid m_1$, then show that $\mathbf{S}_2 \times \mathbf{S}_1 = \mathbf{S}_1$.

Answer: Note: you should assume that all the cryptosystems in this question have equiprobable keys.

Suppose that $\mathbf{S}_2$ has keyword

$$K = (k_1, \dots, k_{m_2})$$

and $\mathbf{S}_1$ has keyword

$$J = (j_1, \dots, j_{m_1}).$$

Then $\mathbf{S}_2 \times \mathbf{S}_1$ has keyword

$$L = (j_1+k_1, \dots, j_{m_2}+k_{m_2}, j_{m_2+1}+k_1, \dots, j_{2m_2}+k_{m_2}, \dots, j_{m_1}+k_{m_2}).$$

Clearly this is a keyword of length m_1.

It remains to show that the probability of each keyword L of length m_1 occurring in the product cipher is $1/26^{m_1}$. This is not difficult, and it is based on the following observation: for any $L = (l_1, \dots, l_{m_1})$ and any $K = (k_1, \dots, k_{m_2})$, there exist a unique J such that $e_{K,J} = e_L$, namely

$$J = (l_1-k_1, \dots, l_{m_2}-k_{m_2}, l_{m_2+1}-k_1, \dots, l_{2m_2}-k_{m_2}, \dots, l_{m_1}-k_{m_2}).$$

From this, the desired result follows easily.

(b) One might try to generalize the previous result by conjecturing that $\mathbf{S}_2 \times \mathbf{S}_1 = \mathbf{S}_3$, where $\mathbf{S}_3$ is the *Vigenère Cipher* with keyword length $\mathrm{lcm}(m_1, m_2)$. Prove that this conjecture is false.

HINT If $m_1 \not\equiv 0 \pmod{m_2}$, then the number of keys in the product cryptosystem $\mathbf{S}_2 \times \mathbf{S}_1$ is less than the number of keys in $\mathbf{S}_3$.

Answer: The product cipher $\mathbf{S}_2 \times \mathbf{S}_1$ has $26^{m_1+m_2}$ keys. However, $\mathbf{S}_3$ has $26^{\mathrm{lcm}(m_1,m_2)}$ keys. We have that $m_1 + m_2 < 2m_1$ because $m_1 > m_2$. Also, $\mathrm{lcm}(m_1, m_2) \geq 2m_1$ because $m_1 \not\equiv 0 \pmod{m_2}$. Therefore $m_1 + m_2 < \mathrm{lcm}(m_1, m_2)$, which completes the proof (following the hint).

3

Block Ciphers and the Advanced Encryption Standard

Exercises

3.1 Let y be the output of Algorithm 3.1 on input x, where π_S and π_P are defined as in Example 3.1. In other words,

$$y = \mathsf{SPN}\left(x, \pi_S, \pi_P, (K^1, \ldots, K^{\mathrm{Nr}+1})\right),$$

where $(K^1, \ldots, K^{\mathrm{Nr}+1})$ is the key schedule. Find a substitution π_{S^*} and a permutation π_{P^*} such that

$$x = \mathsf{SPN}\left(y, \pi_{S^*}, \pi_{P^*}, (K^{\mathrm{Nr}+1}, \ldots, K^1)\right).$$

Answer: Note: Each of the round keys in the decryption algorithm must be permuted in a suitable way.

The decryption algorithm is as follows:

$$x = \mathsf{SPN}\left(y, (\pi_S)^{-1}, (\pi_P)^{-1}, ((\pi_P)^{-1}(K^{\mathrm{Nr}+1}), \ldots, (\pi_P)^{-1}(K^1))\right).$$

3.2 Prove that decryption in a Feistel cipher can be done by applying the encryption algorithm to the ciphertext, with the key schedule reversed.

Answer: *DES* encryption proceeds as follows:

$$\begin{aligned}
&L^0 R^0 = \mathsf{IP}(x)\\
&L^1 = R^0\\
&R^1 = L^0 \oplus f(R^0, K^1)\\
&L^2 = R^1\\
&R^2 = L^1 \oplus f(R^1, K^2)\\
&\vdots\\
&L^{15} = R^{14}\\
&R^{15} = L^{14} \oplus f(R^{14}, K^{15})\\
&L^{16} = R^{15}\\
&R^{16} = L^{15} \oplus f(R^{15}, K^{16})\\
&y = \mathsf{IP}^{-1}(R^{16} L^{16})
\end{aligned}$$

Now, we proceed to decrypt the ciphertext y in a step-by-step fashion. We use prime markings ($'$) to denote the left and right halves of the partially decrypted ciphertext:

$$
\begin{array}{l}
(L')^0(R')^0 = \mathsf{IP}(y) = R^{16}L^{16} \\
(L')^1 = (R')^0 = L^{16} = R^{15} \\
(R')^1 = (L')^0 \oplus f((R')^0, K^{16}) = R^{16} \oplus f(R^{15}, K^{16}) = L^{15} \\
(L')^2 = (R')^1 = L^{15} = R^{14} \\
(R')^2 = (L')^1 \oplus f((R')^1, K^{15}) = R^{15} \oplus f(R^{14}, K^{15}) = L^{14} \\
\vdots \\
(L')^{15} = (R')^{14} = L^2 = R^1 \\
(R')^{15} = (L')^{14} \oplus f((R')^{14}, K^2) = R^2 \oplus f(R^1, K^2) = L^1 \\
(L')^{16} = (R')^{15} = L^1 = R^0 \\
(R')^{16} = (L')^{15} \oplus f((R')^{15}, K^1) = R^1 \oplus f(R^0, K^1) = L^0 \\
y = \mathsf{IP}^{-1}((R')^{16}(L')^{16}) = \mathsf{IP}^{-1}(L^0R^0) = x.
\end{array}
$$

In general, we have $(L')^j = R^{16-j}$ and $(R')^j = L^{16-j}$ for $0 \leq j \leq 16$. This can be proven formally by induction, if desired.

3.3 Let *DES*(x, K) represent the encryption of plaintext x with key K using the *DES* cryptosystem. Suppose $y =$ *DES*(x, K) and $y' =$ *DES*$(c(x), c(K))$, where $c(\cdot)$ denotes the bitwise complement of its argument. Prove that $y' = c(y)$ (i.e., if we complement the plaintext and the key, then the ciphertext is also complemented). Note that this can be proved using only the "high-level" description of *DES* — the actual structure of S-boxes and other components of the system are irrelevant.

Answer: The key fact is that $f(c(A), c(J)) = f(A, J)$, which is easily seen from the description of f. Then, as usual, let the partial encryptions of *DES*(x, K) be denoted L^jR^j, $0 \leq j \leq 16$. Then it is easy to see that the partial encryptions of *DES*$(c(x), c(K))$ are $c(L^j)c(R^j)$, $0 \leq j \leq 16$. This can be proven formally by induction, if desired.

3.4 Before the *AES* was developed, it was suggested to increase the security of *DES* by using the product cipher *DES* × *DES*, as discussed in Section 2.7. This product cipher uses two 56-bit keys.

This exercise considers known-plaintext attacks on product ciphers. In general, suppose that we take the product of any endomorphic cipher $\mathbf{S} = (\mathcal{P}, \mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with itself. Further, suppose that $\mathcal{K} = \{0,1\}^n$ and $\mathcal{P} = \{0,1\}^m$.

Now, assume we have several plaintext-ciphertext pairs for the product cipher $\mathbf{S}^2$, say $(x_1, y_1), \ldots, (x_\ell, y_\ell)$, all of which are obtained using the same unknown key, (K_1, K_2).

(a) Prove that $e_{K_1}(x_i) = d_{K_2}(y_i)$ for all i, $1 \leq i \leq \ell$. Give a heuristic argument that the expected number of keys (K_1, K_2) such that $e_{K_1}(x_i) = d_{K_2}(y_i)$ for all i, $1 \leq i \leq \ell$, is roughly $2^{2n-\ell m}$.

Answer: For $1 \leq i \leq \ell$, we have that $y_i = e_{K_2}(e_{K_1}(x_i))$. Denote $z_i = e_{K_1}(x_i)$. Then $y_i = e_{K_2}(z_i)$, so $e_{K_1}(x_i) = z_i = d_{K_2}(y_i)$, as desired.

Suppose we fix x and choose K_1 at random. If $n \geq m$, then it seems reasonable to hypothesisze that $\mathbf{Pr}[e_{K_1}(x) = z] = 2^{-m}$ for all z. Similarly, if we fix y and choose K_2 at random, it seems reasonable to hypothesisze that $\mathbf{Pr}[d_{K_2}(y) = z] = 2^{-m}$ for all z. Therefore, for fixed x and y, we would estimate that $\mathbf{Pr}[d_{K_2}(y) = e_{K_1}(x)] = 2^{-m}$.

Now given $x_1, \ldots x_\ell$ and $y_1, \ldots, y_\ell$, we would estimate (assuming independence) that

$$\mathbf{Pr}[d_{K_2}(y_i) = e_{K_1}(x_i), 1 \leq i \leq \ell] = 2^{-\ell m}.$$

Since there are 2^{2n} possible pairs (K_1, K_2), the expected number or pairs that satisfy the given conditions is $2^{2n} \times 2^{-\ell m} = 2^{2n-\ell m}$. (Note that this is a heuristic estimate, and not a proof.)

(b) Assume that $\ell \geq 2n/m$. A time-memory trade-off can be used to compute the unknown key (K_1, K_2). We compute two lists, each containing 2^n items, where each item contains an ℓ-tuple of elements of $\mathcal{P}$ as well as an element of $\mathcal{K}$. If the two lists are sorted, then a common ℓ-tuple can be identified by means of a linear search through each of the two lists. Show that this algorithm requires $2^{n+m+1}\ell + 2^{2n+1}$ bits of memory and $\ell 2^{n+1}$ encryptions and/or decryptions.

Answer: Note that the storage requirement is $2^{n+1}(\ell m + n)$ bits.

Suppose elements $x_1, \ldots x_\ell$ and $y_1, \ldots, y_\ell$ are given, where $e_{K_1}(x_i) = d_{K_2}(y_i)$ for all i, $1 \leq i \leq \ell$. We are trying to determine the pair (K_1, K_2).

For every binary n-tuple, K, we construct the tuple

$$(e_K(x_1), \ldots, e_K(x_\ell), K).$$

Call the resulting list of 2^n tuples $\mathcal{L}_1$. Then, for every binary n-tuple, K, we construct the tuple

$$(d_K(y_1), \ldots, d_K(y_\ell), K).$$

Call the resulting list of 2^n tuples $\mathcal{L}_2$.

It takes $\ell\, 2^n$ encryptions to construct $\mathcal{L}_1$, and $\ell\, 2^n$ decryptions to construct $\mathcal{L}_2$. Each tuple in $\mathcal{L}_1$ and $\mathcal{L}_2$ requires $\ell m + n$ bits of storage, so the total storage requirement for $\mathcal{L}_1$ and $\mathcal{L}_2$ is $2^{n+1}(\ell m + n)$ bits.

We can sort the $\mathcal{L}_1$ and $\mathcal{L}_2$ lexicographically by the values of the first ℓ co-ordinates of each tuple. Then we can easily identify all tuples

$$(a_1, \ldots, a_\ell, A) \in \mathcal{L}_1$$

and

$$(b_1, \ldots, b_\ell, B) \in \mathcal{L}_2$$

such that $a_i = b_i$ for $1 \leq i \leq \ell$. This will happen when $A = K_1$ and $B = K_2$, but it may happen for other pairs (A, B) as well. However, we argued in part (a) that the expected number of pairs for which we find a "match" is $2^{2n-\ell m}$. We are now assuming that $2n \leq \ell m$, so $2^{2n-\ell m} \leq 1$ and we do not expect many matches to occur. (Hopefully, there is only one match, the correct one.)

(c) Show that the memory requirement of the attack can be reduced by a factor of 2^t if the total number of encryptions is increased by a factor of 2^t.

HINT Break the problem up into 2^{2t} subcases, each of which is specified by simultaneously fixing t bits of K_1 and t bits of K_2.

Answer: Suppose that I_1 and I_2 are binary t-tuples (note that there are 2^{2t} choices for the pair (I_1, I_2)). For a given pair (I_1, I_2), we can construct the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ in which we require that the last t bits of each K in $\mathcal{L}_1$ are specified by I_1, and the last t bits of each K in $\mathcal{L}_2$ are specified by I_2. This reduces the memory requirement of each list by a factor of 2^t, and the time required to construct $\mathcal{L}_1$ and $\mathcal{L}_2$ (for a given pair (I_1, I_2)) is also reduced by a factor of 2^t.

We search for a match in $\mathcal{L}_1$ and $\mathcal{L}_2$ exactly as before. However, we now have to repeat this for every possible pair (I_1, I_2) in order to be guaranteed that we will find a match. We have 2^{2t} cases to consider, each of which is faster by a factor of 2^t. The total time is therefore increased by a factor of 2^t.

3.5 Suppose that we have the following 128-bit *AES* key, given in hexadecimal notation:

`2B7E151628AED2A6ABF7158809CF4F3C`

Construct the complete key schedule arising from this key.

Answer: This example is worked out in detail, starting on page 27 of the official FIPS 197 description, which can be found at the following web page:

`csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

3.6 Compute the encryption of the following plaintext (given in hexadecimal notation) using the 10-round *AES*:

`3243F6A8885A308D313198A2E0370734`

Use the 128-bit key from the previous exercise.

Answer: This example is worked out in detail, starting on page 33 of the official FIPS 197 description, which can be found at the following web page:

`csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

3.7 Suppose a sequence of plaintext blocks, $x_1 \cdots x_n$, yields the ciphertext sequence $y_1 \cdots y_n$. Suppose that one ciphertext block, say y_i, is transmitted incorrectly (i.e., some 1's are changed to 0's and vice versa). Show that the number of plaintext blocks that will be decrypted incorrectly is equal to one if ECB or OFB modes are used for encryption; and equal to two if CBC or CFB modes are used.

Answer: It is immediate that there is only one incorrectly decrypted ciphertext block when ECB or OFB modes are used for encryption.

Suppose that CBC mode is used, and the ciphertext block y_i is transmitted incorrectly as y_i^*. $x_1, \ldots, x_{i-1}$ are decrypted correctly. The next two ciphertext blocks are decrypted incorrectly:

$$\begin{aligned} x_i^* &= d_K(y_i^*) \oplus y_{i-1} \quad \text{and} \\ x_{i+1}^* &= d_K(y_{i+1}) \oplus y_i^*. \end{aligned}$$

Then all subsequent ciphertext blocks are decrypted correctly.

Suppose that CFB mode is used, and the ciphertext block y_i is transmitted incorrectly as y_i^*. $x_1, \ldots, x_{i-1}$ are decrypted correctly. The next two ciphertext blocks are decrypted incorrectly:

$$\begin{aligned} x_i^* &= e_K(y_{i-1}) \oplus y_i^* \quad \text{and} \\ x_{i+1}^* &= e_K(y_i^*) \oplus y_{i+1}. \end{aligned}$$

Then all subsequent ciphertext blocks are decrypted correctly.

3.8 The purpose of this question is to investigate a time-memory trade-off for a chosen plaintext attack on a certain type of cipher. Suppose we have a cryptosystem in which $\mathcal{P} = \mathcal{C} = \mathcal{K}$, which attains perfect secrecy. Then it must be the case that

$e_K(x) = e_{K_1}(x)$ implies $K = K_1$. Denote $\mathcal{P} = Y = \{y_1, \ldots, y_N\}$. Let x be a fixed plaintext. Define the function $g : Y \rightarrow Y$ by the rule $g(y) = e_y(x)$. Define a directed graph G having vertex set Y, in which the edge set consists of all the directed edges of the form $(y_i, g(y_i))$, $1 \leq i \leq N$.

```
Algorithm 3.1: TIME-MEMORY TRADE-OFF($x$)

$y_0 \leftarrow y$
$backup \leftarrow$ false
while $g(y) \neq y_0$
  do
    if $y = z_j$ for some $j$ and not $backup$
      then
        $y \leftarrow g^{-T}(z_j)$
        $backup \leftarrow$ true
      else
        $y \leftarrow g(y)$
        $K \leftarrow y$
```

(a) Prove that G consists of the union of disjoint directed cycles.
Answer: $g(y) = g(y')$ implies $e_y(x) = e_{y'}(x)$, which implies $y = y'$ (as remarked above). Therefore g is a permutation of the set Y, and its representation as a directed graph is a union of disjoint directed cycles.

(b) Let T be a desired time parameter. Suppose we have a set of elements $Z = \{z_1, \ldots, z_m\} \subseteq Y$ such that, for every element $y_i \in Y$, either y_i is contained in a cycle of length at most T, or there exists an element $z_j \neq y_i$ such that the distance from y_i to z_j (in G) is at most T. Prove that there exists such a set Z such that

$$|Z| \leq \frac{2N}{T},$$

so $|Z|$ is $O(N/T)$.
Answer: Let the cycles in T be denoted $C_1, C_2, \ldots, C_r$. Note that $\sum |C_i| = N$. It is easy to construct a set Z, satisfying the desired properties, such that every cycle C_i contains exactly $\left\lceil \frac{|C_i|}{T} \right\rceil$ points of Z. It can be verified that $\lceil x \rceil \leq 2x$ for all $x \geq 1$. Hence we have that

$$\sum_{i=1}^{r} \left\lceil \frac{|C_i|}{T} \right\rceil \leq \sum_{i=1}^{r} \frac{2|C_i|}{T} = \frac{2N}{T}.$$

(c) For each $z_j \in Z$, define $g^{-T}(z_j)$ to be the element y_i such that $g^T(y_i) = z_j$, where g^T is the function that consists of T iterations of g. Construct a table X consisting of the ordered pairs $(z_j, g^{-T}(z_j))$, sorted with respect to their first coordinates.

A pseudo-code description of an algorithm to find K, given $y = e_K(x)$, is presented. Prove that this algorithm finds K in at most T steps. (Hence the time-memory trade-off is $O(N)$.)
Answer: Note: The input to this algorithm should be y rather than x.

The algorithm requires at most T iterations of the while loop to find $y = z_j$, and then at most T further iterations until $g(y) = y_0$. Therefore the total number of iterations is $O(T)$. Each iteration requires time $O(1) + O(\log N)$ (assuming we do a binary search of the z_j's), so the total time is $O(T \log N)$.

The memory requirement is $O(N \log N/T)$ bits. Therefore the product of time and memory is $O(N(\log N)^2)$. If we ignore the logarithmic factor (as is usually done in analyses of this type), the product is $O(N)$.

(d) Describe a pseudo-code algorithm to construct the desired set Z in time $O(NT)$ without using an array of size N.

Answer: We construct Z, as well as the set X of ordered pairs of the form $(z, g^{-T}(z))$, as follows:

```
Algorithm: CONSTRUCTXANDZ($x$)

$Z \leftarrow \emptyset$
for $i \leftarrow 1$ to $N$
  do
    $newcycle \leftarrow true$
    $y_0 \leftarrow y_i$
    $y \leftarrow y_i$
    for $j \leftarrow 1$ to $T$
      do
        $y_0 \leftarrow g(y_0)$
        if $y_0 \in Z$
          then $newcycle \leftarrow false$
    if $newcycle$
      then
        $endcycle \leftarrow false$
        while not $endcycle$
          do
            $Z \leftarrow Z \cup \{y_0\}$
            $X \leftarrow X \cup \{(y_0, y)\}$
            $y \leftarrow y_0$
            for $j \leftarrow 1$ to $T$
              do
                $y_0 \leftarrow g(y_0)$
                if $y_0 \in Z$
                  then $endcycle \leftarrow true$
```

3.9 Suppose that $\mathbf{X_1}$, $\mathbf{X_2}$ and $\mathbf{X_3}$ are independent discrete random variables defined on the set $\{0, 1\}$. Let ϵ_i denote the bias of $\mathbf{X_i}$, for $i = 1, 2, 3$. Prove that $\mathbf{X_1} \oplus \mathbf{X_2}$ and $\mathbf{X_2} \oplus \mathbf{X_3}$ are independent if and only if $\epsilon_1 = 0$, $\epsilon_3 = 0$ or $\epsilon_2 = \pm 1/2$.

Answer: $\mathbf{X_1} \oplus \mathbf{X_2}$ has bias $2\epsilon_1\epsilon_2$ and $\mathbf{X_2} \oplus \mathbf{X_3}$ has bias $2\epsilon_2\epsilon_3$. Suppose that $\mathbf{X_1} \oplus \mathbf{X_2}$ and $\mathbf{X_2} \oplus \mathbf{X_3}$ are independent. Then the bias of $(\mathbf{X_1} \oplus \mathbf{X_2}) \oplus (\mathbf{X_2} \oplus \mathbf{X_3})$ would be $2(2\epsilon_1\epsilon_2)(2\epsilon_2\epsilon_3)$. However,

$$(\mathbf{X_1} \oplus \mathbf{X_2}) \oplus (\mathbf{X_2} \oplus \mathbf{X_3}) = \mathbf{X_1} \oplus \mathbf{X_3}$$

has bias $2\epsilon_1\epsilon_3$. Therefore

$$4\epsilon_1(\epsilon_2)^2\epsilon_3 = \epsilon_1\epsilon_3.$$

This implies that $\epsilon_1 = 0$, $\epsilon_3 = 0$ or $\epsilon_2 = \pm 1/2$.

Conversely, suppose that $\epsilon_1 = 0$, $\epsilon_3 = 0$ or $\epsilon_2 = \pm 1/2$. The two random variables $\mathbf{X_1} \oplus \mathbf{X_2}$ and $\mathbf{X_2} \oplus \mathbf{X_2}$ are independent if and only if

$$\mathbf{Pr}[(\mathbf{X_1} \oplus \mathbf{X_2} = a) \text{ and } (\mathbf{X_2} \oplus \mathbf{X_3} = b)] = \mathbf{Pr}[\mathbf{X_1} \oplus \mathbf{X_2} = a] \times \mathbf{Pr}[\mathbf{X_2} \oplus \mathbf{X_3} = b]$$

for $a, b \in \{0, 1\}$. These four conditions are as follows:

$$\begin{aligned}
&p_1 p_2 p_3 + (1-p_1)(1-p_2)(1-p_3)\\
&\quad = (p_1 p_2 + (1-p_1)(1-p_2))(p_2 p_3 + (1-p_2)(1-p_3)),\\
&p_1 p_2 (1-p_3) + (1-p_1)(1-p_2)p_3\\
&\quad = (p_1 p_2 + (1-p_1)(1-p_2))(p_2(1-p_3) + (1-p_2)p_3),\\
&p_1(1-p_2)(1-p_3) + (1-p_1)p_2 p_3\\
&\quad = (p_1(1-p_2) + (1-p_1)p_2)(p_2 p_3 + (1-p_2)(1-p_3)), \quad \text{and}\\
&p_1(1-p_2)p_3 + (1-p_1)p_2(1-p_3)\\
&\quad = (p_1(1-p_2) + (1-p_1)p_2)(p_2(1-p_3) + (1-p_2)p_3).
\end{aligned}$$

It is straightforward to verify that these four conditions are satisfied when $p_1 = 1/2$, when $p_3 = 1/2$, when $p_2 = 0$ and when $p_2 = 1$.

3.10 For the each of eight *DES* S-boxes, compute the bias of the random variable

$$\mathbf{X_2} \oplus \mathbf{Y_1} \oplus \mathbf{Y_2} \oplus \mathbf{Y_3} \oplus \mathbf{Y_4}.$$

(Note that these biases are all relatively large in absolute value.)
Answer: The biases for $S_1, \dots, S_8$ are (respectively)

$$-\frac{18}{64}, -\frac{12}{64}, \frac{10}{64}, \frac{8}{64}, -\frac{20}{64}, \frac{10}{64}, -\frac{14}{64}, \text{ and } -\frac{16}{64}.$$

3.11 The *DES* S-box S_4 has some unusual properties:

(a) Prove that the second row of S_4 can be obtained from the first row by means of the following mapping:

$$(y_1, y_2, y_3, y_4) \mapsto (y_2, y_1, y_4, y_3) \oplus (0, 1, 1, 0),$$

where the entries are represented as binary strings.
Answer: This is a straghtforward verification.

(b) Show that any row of S_4 can be transformed into any other row by a similar type of operation.
Answer: The third row can be transformed into the fourth row by the same mapping used in part (a).

To transform the first row (row $(0, 0)$) into the fourth row (row $(1, 1)$), the following operations are performed.

i. Let the entry in column (c_1, c_2, c_3, c_4) of row $(0, 0)$ be (y_1, y_2, y_3, y_4), where all vectors have entries 0 and 1.

ii. Compute $(y_4, y_3, y_2, y_1) \oplus (0, 1, 1, 0)$.

iii. The result is the entry in column $(c_1, c_2, c_3, c_4) \oplus (0, 1, 1, 1)$ of row $(1, 1)$.

By composing these transformations, any row of S_4 can be transformed into any other row.

3.12 Suppose that $\pi_S : \{0, 1\}^m \to \{0, 1\}^n$ is an S-box. Prove the following facts about the function N_L.

(a) $N_L(0, 0) = 2^m$.
Answer: This is trivial.

(b) $N_L(a,0) = 2^m - 1$ for all integers a such that $0 \leq a \leq 2^m - 1$.
Answer: Note: This should read "$N_L(a,0) = 2^{m-1}$ for all integers a such that $0 < a \leq 2^m - 1$".

For $a \in \{0,1\}^m$, $a \neq (0,\dots,0)$, there are exactly 2^{m-1} bitstrings $x \in \{0,1\}^m$ such that $\sum a_i x_i \equiv 0 \pmod 2$.

(c) For all integers a such that $0 \leq a \leq 2^m - 1$, it holds that

$$\sum_{a=0}^{2^m-1} N_L(a,b) = 2^{2m-1} \pm 2^{m-1}.$$

Answer: Note: The first line should read "For all integers b such that $0 \leq b \leq 2^m - 1$,".

Suppose x is fixed; then $y = \pi_S(x)$ and $c = \sum b_i y_i \bmod 2$ is determined. If $x \neq 0$, then there are 2^{m-1} choices for a such that $\sum a_i x_i \bmod 2 = c$ (by part (b)). If $x = 0$, then there are either 0 or 2^m choices for a such that $\sum a_i x_i \bmod 2 = c$ (by part (a), depending on whether $c = 0$ or 1, respectively). Therefore it follows that

$$\sum_{a=0}^{2^m-1} N_L(a,b) = (2^m - 1)2^{m-1} + (0 \text{ or } 2^m) = 2^{2m-1} \pm 2^{m-1}.$$

(d) It holds that

$$\sum_{a=0}^{2^m-1} \sum_{b=0}^{2^n-1} N_L(a,b) \in \{2^{n+2m-1}, 2^{n+2m-1} + 2^{n+m-1}\}.$$

Answer: If $x \neq 0$, then there are 2^{m-1} choices for a for each b (by part (c)). Therefore we obtain $2^{n+m-1}(2^m - 1)$ quadruples (a,b,x,y) with $x \neq 0$ such that $\sum a_i x_i + \sum b_i y_i \bmod 2 = 0$.

Now we consider $x = 0$. Define $y_0 = \pi_S(0,\dots,0)$. If $y_0 = 0$, then all possible a and b work, so the number of quadruples $(a,b,x = 0, y = 0)$ is 2^{n+m}. If $y_0 \neq 0$, then for each a, there are 2^{n-1} choices for b, and the number of quadruples $(a,b,x = 0, y = y_0)$ is 2^{n+m-1}.

In total, the number of quadruples is

$$2^{n+m-1}(2^m-1)+(2^{n+m} \text{ or } 2^{n+m-1}) = 2^{n+2m-1} \text{ or } 2^{n+2m-1}+2^{n+m-1}.$$

3.13 An S-box $\pi_S : \{0,1\}^m \to \{0,1\}^n$ is said to be *balanced* if

$$|{\pi_S}^{-1}(y)| = 2^{n-m}$$

for all $y \in \{0,1\}^n$. Prove the following facts about the function N_L for a balanced S-box.

(a) $N_L(0,b) = 2^m - 1$ for all integers b such that $0 \leq b \leq 2^n - 1$.
Answer: Note: This should read "$N_L(0,b) = 2^{m-1}$ for all integers b such that $0 < b \leq 2^n - 1$".

When $b \neq 0$, there are 2^{n-1} y's such that $\sum b_i y_i \bmod 2 = 0$. For each such y, there are exactly 2^{n-m} x's such that $\pi_S(x) = y$. Therefore, $N_L(0,b) = 2^{n-1} \times 2^{n-m} = 2^{m-1}$.

(b) For all integers a such that $0 \leq a \leq 2^m - 1$, it holds that

$$\sum_{b=0}^{2^n-1} N_L(a,b) = 2^{m+n-1} - 2^{m-1} + i2^n,$$

where i is an integer such that $0 \leq i \leq 2^{m-n}$.
Answer: When $y \neq 0$, thre are 2^{m-n} x's such that $y = \pi_S(x)$. For each such x, there are 2^{n-1} b's such that $\sum b_i y_i \equiv \sum a_i x_i \bmod 2$. Thus we obtain $2^{m-1}(2^n - 1)$ triples (b, x, y) with $y \neq 0$ such that $\sum a_i x_i + \sum b_i y_i \bmod 2 = 0$.

Now consider $y = 0$. Define

$$X_0 = \{x \in {\pi_S}^{-1}(0) : \sum a_i x_i \bmod 2 = 0\}$$

and denote $i = |X_0|$. Note that $0 \leq i \leq 2^{n-m}$. For each $x \in X_0$ and for every b, it holds that $\sum a_i x_i + \sum b_i y_i \bmod 2 = 0$. on the other hand, if $x \in {\pi_S}^{-1}(0) \backslash X_0$, then the condition holds for no b. Hence, we get $i2^{\,n}$ triples $(b, x, y = 0)$ with $y = 0$ such that $\sum a_i x_i + \sum b_i y_i \bmod 2 = 0$. Hence, the total number of triples is $2^{m+n-1} - 2^{m-1} + i2^n$, where $0 \leq i \leq 2^{n-m}$.

3.14 Suppose that the S-box of Example 3.1 is replaced by the S-box defined by the following substitution $\pi_{S'}$:

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_{S'}(z)$	8	4	2	1	C	6	3	D	A	5	E	7	F	B	9	0

(a) Compute the table of values N_L for this S-box.
Answer: The table is as follows:

	b															
a	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	10	6	8	10	8	8	6	4	6	6	8	10	8	4	10
2	8	10	8	10	6	8	6	8	6	8	10	4	4	6	8	10
3	8	8	10	10	8	12	10	6	6	6	8	8	10	6	12	8
4	8	10	8	6	8	10	8	6	10	4	10	8	6	8	6	4
5	8	12	6	6	10	10	8	12	6	10	8	8	8	8	10	6
6	8	8	12	8	10	10	6	10	8	8	8	12	6	6	6	10
7	8	6	6	8	12	6	10	8	8	6	6	8	4	6	10	8
8	8	10	10	8	8	6	6	8	10	8	4	6	10	4	8	6
9	8	8	8	12	10	10	6	10	10	6	6	6	8	12	8	8
A	8	12	10	10	6	6	12	8	8	8	6	10	6	10	8	8
B	8	6	12	6	8	6	8	10	4	6	8	6	8	10	8	6
C	8	8	10	10	12	8	10	6	8	12	10	6	8	8	6	6
D	8	6	8	6	6	12	10	8	8	10	4	6	6	8	6	8
E	8	6	6	12	6	8	8	10	6	8	8	10	8	6	6	4
F	8	8	8	8	8	8	12	12	10	6	10	6	10	6	6	10

(b) Find a linear approximation using three active S-boxes, and use the piling-up lemma to estimate the bias of the random variable

$$\mathbf{X_{16}} \oplus \mathbf{U_1^4} \oplus \mathbf{U_9^4}.$$

Answer: The approximation incorporates the following three active S-boxes:

- In S_4^1, the random variable $\mathbf{T_1} = \mathbf{U_{16}^1} \oplus \mathbf{V_{13}^1}$ has bias $-1/4$
- In S_1^2, the random variable $\mathbf{T_2} = \mathbf{U_4^2} \oplus \mathbf{V_1^2}$ has bias $-1/4$
- In S_1^3, the random variable $\mathbf{T_3} = \mathbf{U_1^3} \oplus \mathbf{V_1^3} \oplus \mathbf{V_3^3}$ has bias $-1/4$

Using the piling-up lemma, the bias of the random variable $\mathbf{T_1} \oplus \mathbf{T_2} \oplus \mathbf{T_3}$ is estimated to be $-1/16$. Now, use the following relations:

$$\begin{aligned}
\mathbf{U^1_{16}} &= \mathbf{X_{16}} \oplus \mathbf{K^1_{16}}\\
\mathbf{U^2_4} &= \mathbf{V^1_{13}} \oplus \mathbf{K^2_4}\\
\mathbf{U^3_1} &= \mathbf{V^2_1} \oplus \mathbf{K^3_1}\\
\mathbf{U^4_1} &= \mathbf{V^3_1} \oplus \mathbf{K^4_1}\\
\mathbf{U^4_9} &= \mathbf{V^3_3} \oplus \mathbf{K^4_9}
\end{aligned}$$

to show that

$$\mathbf{T_1} \oplus \mathbf{T_2} \oplus \mathbf{T_3} = \mathbf{X_{16}} \oplus \mathbf{U^4_1} \oplus \mathbf{U^4_9} \oplus \text{key bits.}$$

Therefore we estimate that the bias of $\mathbf{X_{16}} \oplus \mathbf{U^4_1} \oplus \mathbf{U^4_9}$ is $\pm 1/16$.

(c) Describe a linear attack, analogous to Algorithm 3.2, that will find eight subkey bits in the last round.
Answer: The algorithm is as follows:

```
Algorithm: LINEARATTACK($\mathcal{T}$, $T$, ${\pi_{S'}}^{-1}$)

for $(L_1, L_2) \leftarrow (0,0)$ to $(F,F)$
  do $Count[L_1, L_2] \leftarrow 0$
for each $(x, y) \in \mathcal{T}$
  do { for $(L_1, L_2) \leftarrow (0,0)$ to $(F,F)$
         do { $v^4_{(1)} \leftarrow L_1 \oplus y_{(1)}$
              $v^4_{(3)} \leftarrow L_2 \oplus y_{(3)}$
              $u^4_{(1)} \leftarrow {\pi_{S'}}^{-1}(v^4_{(1)})$
              $u^4_{(3)} \leftarrow {\pi_{S'}}^{-1}(v^4_{(3)})$
              $z \leftarrow x_{16} \oplus u^4_1 \oplus u^4_9$
              if $z = 0$
                then $Count[L_1, L_2] \leftarrow Count[L_1, L_2] + 1$
$max \leftarrow -1$
for $(L_1, L_2) \leftarrow (0,0)$ to $(F,F)$
  do { $Count[L_1, L_2] \leftarrow |Count[L_1, L_2] - T/2|$
       if $Count[L_1, L_2] > max$
         then { $max \leftarrow Count[L_1, L_2]$
                $maxkey \leftarrow (L_1, L_2)$
output $(maxkey)$
```

(d) Implement your attack and test it to see how many plaintexts are required in order for the algorithm to find the correct subkey bits (approximately 1000–1500 plaintexts should suffice; this attack is more efficient than Algorithm 3.2 because the bias is larger by a factor of 2, which means that the number of plaintexts can be reduced by a factor of about 4).

3.15 Suppose that the S-box of Example 3.1 is replaced by the S-box defined by the following substitution $\pi_{S''}$:

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_{S''}(z)$	E	2	1	3	D	9	0	6	F	4	5	A	8	C	7	B

(a) Compute the table of values N_D for this S-box.
Answer: The table of values is as follows:

a'	b'															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	2	0	4	0	2	0	0	0	0	2	4	0	0	2
2	0	2	0	0	0	0	0	2	0	0	2	0	0	2	2	6
3	0	2	0	4	0	2	0	0	0	2	0	4	0	2	0	0
4	0	4	2	2	0	2	0	2	2	0	0	2	0	0	0	0
5	0	0	0	4	0	0	0	4	0	0	0	0	2	2	2	2
6	0	0	0	0	2	0	2	0	2	0	2	0	2	2	2	2
7	0	0	4	2	2	0	0	0	4	2	0	0	0	0	2	0
8	0	2	0	0	2	4	2	2	0	2	0	0	0	2	0	0
9	0	6	0	0	0	0	2	0	0	0	2	4	0	2	0	0
A	0	0	2	0	0	0	0	2	4	0	4	2	0	0	2	0
B	0	0	0	0	2	2	2	2	0	0	0	0	4	0	4	0
C	0	0	2	0	0	2	4	0	2	0	0	0	2	2	2	0
D	0	0	2	2	2	0	2	0	0	2	6	0	0	0	0	0
E	0	0	2	2	0	0	0	0	2	6	0	0	0	0	0	4
F	0	0	0	0	2	4	0	2	0	2	0	2	2	2	0	0

(b) Find a differential trail using four active S-boxes, namely, S_1^1, S_4^1, S_4^2 and S_4^3, that has propagation ratio $27/2048$.
Answer: The following propagation ratios of differentials can be verified from the table computed in part (a):

- In S_1^1, $R_p(1001, 0001) = 3/8$
- In S_4^1, $R_p(1001, 0001) = 3/8$
- In S_4^2, $R_p(1001, 0001) = 3/8$
- In S_4^3, $R_p(0001, 1100) = 1/4$

These differentials can be combined to form a differential trail of the first three rounds of the SPN:

$$R_p(\texttt{1001 0000 0000 1001}, \texttt{0000 0000 0000 1100}) = \frac{27}{2048}.$$

Hence, it can be verified that

$$x' = \texttt{1001 0000 0000 1001} \Rightarrow (u^4)' = \texttt{0001 0001 0000 0000}$$

with probability $27/2048$.

(c) Describe a differential attack, analogous to Algorithm 3.3, that will find eight subkey bits in the last round.
Answer: The algorithm is as follows:

```
Algorithm: DIFFERENTIALATTACK($\mathcal{T}, T, {\pi_{S''}}^{-1}$)

for $(L_1, L_2) \leftarrow (0, 0)$ to $(F, F)$
  do $Count[L_1, L_2] \leftarrow 0$
for each $(x, y, x^*, y^*) \in \mathcal{T}$
  do if $(y_{(3)} = (y_{(3)})^*)$ and $(y_{(4)} = (y_{(4)})^*)$
     then for $(L_1, L_2) \leftarrow (0, 0)$ to $(F, F)$
            do $v^4_{(1)} \leftarrow L_1 \oplus y_{(1)}$
               $v^4_{(2)} \leftarrow L_2 \oplus y_{(2)}$
               $u^4_{(1)} \leftarrow {\pi_{S''}}^{-1}(v^4_{(1)})$
               $u^4_{(2)} \leftarrow {\pi_{S''}}^{-1}(v^4_{(2)})$
               $(v^4_{(1)})^* \leftarrow L_1 \oplus (y_{(1)})^*$
               $(v^4_{(2)})^* \leftarrow L_2 \oplus (y_{(2)})^*$
               $(u^4_{(1)})^* \leftarrow {\pi_{S''}}^{-1}((v^4_{(1)})^*)$
               $(u^4_{(2)})^* \leftarrow {\pi_{S''}}^{-1}((v^4_{(2)})^*)$
               $(u^4_{(1)})' \leftarrow u^4_{(1)} \oplus (u^4_{(1)})^*$
               $(u^4_{(2)})' \leftarrow u^4_{(2)} \oplus (u^4_{(2)})^*$
               if $((u^4_{(1)})' = 0001)$ and $((u^4_{(2)})' = 0001)$
                 then $Count[L_1, L_2] \leftarrow Count[L_1, L_2] + 1$
$max \leftarrow -1$
for $(L_1, L_2) \leftarrow (0, 0)$ to $(F, F)$
  do if $Count[L_1, L_2] > max$
     then $max \leftarrow Count[L_1, L_2]$
          $maxkey \leftarrow (L_1, L_2)$
output $(maxkey)$
```

(d) Implement your attack and test it to see how many plaintexts are required in order for the algorithm to find the correct subkey bits (approximately 100–200 plaintexts should suffice; this attack is not as efficient as Algorithm 3.3 because the propagation ratio is smaller by a factor of 2).

3.16 Suppose that we use the SPN presented in Example 3.1, but the S-box is replaced by a function π_T that is not a permutation. This means, in particular, that π_T is not surjective. Use this fact to derive a ciphertext-only attack that can be used to determine the key bits in the last round, given a sufficient number of ciphertexts which all have been encrypted using the same key.

Answer: Suppose that ${\pi_T}^{-1}(z) = \emptyset$ for some $z \in \{0,1\}^4$. Suppose we are given a set of ciphertexts $\mathcal{T}$, all of which are encrypted using the same unknown key, K. For each $y = y_{(1)} \parallel y_{(2)} \parallel y_{(3)} \parallel y_{(4)} \in T$, and for each i, $1 \leq i \leq 4$, it must be the case that $y_{(i)} \oplus K_{(i)} \neq z$. For $1 \leq i \leq 4$, define

$$\mathcal{K}_i = \{0,1\}^4 \backslash \{z \oplus y_{(i)} : y \in \mathcal{T}\}.$$

Then $K_{(i)} \in \mathcal{K}_i$, $1 \leq i \leq 4$. If $|\mathcal{T}|$ is reasonably large, then we expect that $|\mathcal{K}_i| = 1$ for $1 \leq i \leq 4$, and hence the key K can be determined.

4

Cryptographic Hash Functions

Exercises

4.1 Suppose $h : \mathcal{X} \rightarrow \mathcal{Y}$ is an (N, M)-hash function. For any $y \in \mathcal{Y}$, let

$$h^{-1}(y) = \{x : h(x) = y\}$$

and denote $s_y = |h^{-1}(y)|$. Define

$$S = |\{\{x_1, x_2\} : h(x_1) = h(x_2)\}|.$$

Note that S counts the number of unordered pairs in $\mathcal{X}$ that collide under h.

(a) Prove that

$$\sum_{y \in \mathcal{Y}} s_y = N,$$

so the mean of the s_y's is

$$\overline{s} = \frac{N}{M}.$$

Answer: Clearly the sets $h^{-1}(y)$, $y \in \mathcal{Y}$, form a partition of $\mathcal{X}$. Hence, $\sum_{y \in \mathcal{Y}} s_y = |\mathcal{X}| = N$. Then, because $|\mathcal{Y}| = M$, it is immediate that the mean of the s_y's is N/M.

(b) Prove that

$$S = \sum_{y \in \mathcal{Y}} \binom{s_y}{2} = \frac{1}{2} \sum_{y \in \mathcal{Y}} {s_y}^2 - \frac{N}{2}.$$

Answer: We have the following:

$$\begin{aligned} \sum_{y \in \mathcal{Y}} \binom{s_y}{2} &= \frac{1}{2} \sum_{y \in \mathcal{Y}} {s_y}^2 - \frac{1}{2} \sum_{y \in \mathcal{Y}} s_y \\ &= \frac{1}{2} \sum_{y \in \mathcal{Y}} {s_y}^2 - \frac{N}{2}, \end{aligned}$$

using the result proven in part (a).

(c) Prove that

$$\sum_{y \in \mathcal{Y}} (s_y - \overline{s})^2 = 2N + N - \frac{N^2}{M}.$$

Answer: Note: the term "$2N$" should be "$2S''$.

We have the following:

$$\begin{aligned}\sum_{y \in \mathcal{Y}} (s_y - \overline{s})^2 &= \sum_{y \in \mathcal{Y}} {s_y}^2 - 2\overline{s} \sum_{y \in \mathcal{Y}} s_y + M\overline{s}^2 \\ &= 2\left(S + \frac{N}{2}\right) - \frac{2N}{M} \times N + M \left(\frac{N}{M}\right)^2 \\ &= 2S + N - \frac{N^2}{M}.\end{aligned}$$

(d) Using the result proved in part (c), prove that

$$S \geq \frac{1}{2}\left(\frac{N^2}{M} - N\right).$$

Further, show that equality is attained if and only if

$$s_y = \frac{N}{M}$$

for every $y \in \mathcal{Y}$.

Answer: Clearly

$$\sum_{y \in \mathcal{Y}} (s_y - \overline{s})^2 \geq 0,$$

and this sum is zero if and only if $s_y = \overline{s}$ for all $y \in \mathcal{Y}$. In other words,

$$0 \leq 2S + N - \frac{N^2}{M},$$

and equality occurs if and only if $s_y = N/M$ for all $y \in \mathcal{Y}$. Finally, note that

$$0 \leq 2S + N - \frac{N^2}{M} \Leftrightarrow S \geq \frac{1}{2}\left(\frac{N^2}{M} - N\right).$$

4.2 As in Exercise 4.1, suppose $h : \mathcal{X} \rightarrow \mathcal{Y}$ is an (N, M)-hash function, and let

$$h^{-1}(y) = \{x : h(x) = y\}$$

for any $y \in \mathcal{Y}$. Let ϵ denote the probability that $h(x_1) = h(x_2)$, where x_1 and x_2 are random (not necessarily distinct) elements of $\mathcal{X}$. Prove that

$$\epsilon \geq \frac{1}{M},$$

with equality if and only if

$$|h^{-1}(y)| = \frac{N}{M}$$

for every $y \in \mathcal{Y}$.

Answer: Define

$$T = |\{(x_1, x_2) : h(x_1) = h(x_2)\}|.$$

Then $T = 2S + N$, where S is defined as in Exercise 4.1. (The term "$+N$" accounts for the collisions where $x_1 = x_2$; and each unordered pair $\{x_1, x_2\}$ with $h(x_1) = h(x_2)$ accounts for two ordered pairs, namely, (x_1, x_2) and (x_2, x_1).) Using the result proven in Exercise 4.1, part (d), we have that

$$\epsilon = \frac{2S + N}{N^2} \geq \frac{2\left(\frac{1}{2}\left(\frac{N^2}{M} - N\right)\right) + N}{N^2} = \frac{1}{M}.$$

Further, equality occurs if and only if $s_y = N/M$ for all $y \in \mathcal{Y}$ (as in Exercise 4.1, part (d)).

4.3 Suppose that $h : \mathcal{X} \rightarrow \mathcal{Y}$ is an (N, M)-hash function, let

$$h^{-1}(y) = \{x : h(x) = y\}$$

and let $s_y = |h^{-1}(y)|$ for any $y \in \mathcal{Y}$. Suppose that we try to solve Preimage for the function h, using Algorithm 4.1, assuming that we have only oracle access for h. For a given $y \in \mathcal{Y}$, suppose that $\mathcal{X}_0$ is chosen to be a random subset of $\mathcal{X}$ having cardinality q.

(a) Prove that the success probability of Algorithm 4.1, given y, is

$$1 - \frac{\binom{N-s_y}{q}}{\binom{N}{q}}.$$

Answer: The total number of subsets $\mathcal{X}_0 \subseteq \mathcal{X}$ such that $|\mathcal{X}_0| = q$ is $\binom{N}{q}$. The number of subsets $\mathcal{X}_0 \subseteq \mathcal{X}$ such that $|\mathcal{X}_0| = q$ and $\mathcal{X}_0 \cap h^{-1}(y) = \emptyset$ is $\binom{N-s_y}{q}$. Therefore the failure probability of Algorithm 4.1 is $\binom{N-s_y}{q}/\binom{N}{q}$, and the result follows.

(b) Prove that the average success probabilty of Algorithm 4.1 (over all $y \in \mathcal{Y}$) is

$$1 - \frac{1}{M}\sum_{y \in \mathcal{Y}} \frac{\binom{N-s_y}{q}}{\binom{N}{q}}.$$

Answer: The average success probability is

$$\frac{1}{M}\sum_{y \in \mathcal{Y}} \left(1 - \frac{\binom{N-s_y}{q}}{\binom{N}{q}}\right) = 1 - \frac{1}{M}\sum_{y \in \mathcal{Y}} \frac{\binom{N-s_y}{q}}{\binom{N}{q}}.$$

(c) In the case $q = 1$, show that the success probability in part (b) is $1/M$.
Answer: We compute as follows:

$$\begin{aligned} 1 - \frac{1}{M}\sum_{y \in \mathcal{Y}} \frac{\binom{N-s_y}{1}}{\binom{N}{1}} &= 1 - \frac{1}{M}\sum_{y \in \mathcal{Y}} \frac{N - s_y}{N} \\ &= 1 - \frac{1}{M}\left(M - \sum_{y \in \mathcal{Y}} \frac{s_y}{N}\right) \\ &= 1 - \frac{1}{M}(M-1) \\ &= \frac{1}{M}, \end{aligned}$$

where we use the fact that $\sum_{y \in \mathcal{Y}} s_y = N$, which was proven in Exercise 4.1(a).

4.4 Suppose that $h : \mathcal{X} \rightarrow \mathcal{Y}$ is an (N, M)-hash function, let

$$h^{-1}(y) = \{x : h(x) = y\}$$

and let $s_y = |h^{-1}(y)|$ for any $y \in \mathcal{Y}$. Suppose that we try to solve Second Preimage for the function h, using Algorithm 4.2, assuming that we have only oracle access for h. For a given $x \in \mathcal{Y}$, suppose that $\mathcal{X}_0$ is chosen to be a random subset of $\mathcal{X}\backslash\{x\}$ having cardinality $q - 1$.

(a) Prove that the success probability of Algorithm 4.2, given x, is

$$1 - \frac{\binom{N-s_y}{q-1}}{\binom{N-1}{q-1}}.$$

Answer: The total number of subsets $\mathcal{X}_0 \subseteq \mathcal{X}\backslash\{x\}$ such that $|\mathcal{X}_0| = q-1$ is $\binom{N-1}{q-1}$. Denote $h(x) = y$; then the number of subsets $\mathcal{X}_0 \subseteq \mathcal{X}\backslash\{x\}$ such that $|\mathcal{X}_0| = q-1$ and $\mathcal{X}_0 \cap h^{-1}(y) = \emptyset$ is $\binom{N-s_y}{q-1}$. Therefore the failure probability of Algorithm 4.2 is $\binom{N-s_y}{q-1}/\binom{N-1}{q-1}$, and the result follows.

(b) Prove that the average success probabilty of Algorithm 4.2 (over all $x \in \mathcal{X}$) is

$$1 - \frac{1}{N}\sum_{y\in\mathcal{Y}} \frac{s_y\binom{N-s_y}{q-1}}{\binom{N-1}{q-1}}.$$

Answer: The average success probability is

$$\frac{1}{N}\sum_{x\in\mathcal{X}}\left(1 - \frac{\binom{N-s_{h(x)}}{q-1}}{\binom{N-1}{q-1}}\right) = 1 - \frac{1}{N}\sum_{y\in\mathcal{Y}} \frac{s_y\binom{N-s_y}{q-1}}{\binom{N-1}{q-1}}.$$

(c) In the case $q = 2$, show that the success probability in part (b) is

$$\frac{\sum_{y\in\mathcal{Y}} {s_y}^2}{N(N-1)} - \frac{1}{N-1}.$$

Answer: We compute as follows:

$$\begin{aligned} 1 - \frac{1}{N}\sum_{y\in\mathcal{Y}} \frac{s_y\binom{N-s_y}{1}}{\binom{N-1}{1}} &= 1 - \frac{1}{N(N-1)}\left(\sum_{y\in\mathcal{Y}} N s_y - \sum_{y\in\mathcal{Y}} {s_y}^2\right)\\ &= 1 - \frac{N}{N-1} + \frac{\sum_{y\in\mathcal{Y}} {s_y}^2}{N(N-1)}\\ &= \frac{\sum_{y\in\mathcal{Y}} {s_y}^2}{N(N-1)} - \frac{1}{N-1},\end{aligned}$$

where we use the fact that $\sum_{y\in\mathcal{Y}} s_y = N$, which was proven in Exercise 4.1(a).

4.5 If we define a hash function (or compression function) h that will hash an n-bit binary string to an m-bit binary string, we can view h as a function from $\mathbb{Z}_{2^n}$ to $\mathbb{Z}_{2^m}$. It is tempting to define h using integer operations modulo 2^m. We show in this exercise that some simple constructions of this type are insecure and should therefore be avoided.

(a) Suppose that $n = m$ and $h : \mathbb{Z}_{2^m} \rightarrow \mathbb{Z}_{2^m}$ is defined as

$$h(x) = x^2 + ax + b \bmod 2^m.$$

Prove that it is easy to solve Second Preimage for any $x \in \mathbb{Z}_{2^m}$ without having to solve a quadratic equation.

Answer: Note: we need to assume that $m \geq 2$.

Suppose that a is even; then $a2^{m-1} \equiv 0 \pmod{2^m}$. Also, $2^{2m-2} \equiv 0 \pmod{2^m}$ because $m \geq 2$. Define $x' = x + 2^{m-1} \bmod 2^m$; then

$$\begin{aligned} h(x') &= (x + 2^{m-1})^2 + a(x + 2^{m-1}) + b \bmod 2^m\\ &= x^2 + 2^m x + 2^{2m-2} + ax + a2^{m-1} + b \bmod 2^m\\ &= x^2 + ax + b \bmod 2^m\\ &= f(x).\end{aligned}$$

Now suppose that a is odd. Define $x' = -x - a \bmod 2^m$; note that $x' \neq x$ because $2x + a$ is odd. Now, we have that

$$\begin{aligned} h(x') &= (-x - a)(-x) + b \bmod 2^m \\ &= (x + a)x + b \bmod 2^m \\ &= h(x). \end{aligned}$$

Therefore, given any x, we can find $x' \neq x$ such that $h(x') = h(x)$.

(b) Suppose that $n > m$ and $h : \mathbb{Z}_{2^n} \rightarrow \mathbb{Z}_{2^m}$ is defined to be a polynomial of degree d:

$$h(x) = \sum_{i=0}^{d} a_i x^i \bmod 2^m,$$

where $a_i \in \mathbb{Z}$ for $0 \leq i \leq d$. Prove that it is easy to solve Second Preimage for any $x \in \mathbb{Z}_{2^n}$ without having to solve a polynomial equation.

Answer: Define $x' = x + 2^m \bmod 2^n$. Then $x' \neq x$ and $h(x') = h(x)$.

4.6 Suppose that $f : \{0,1\}^m \rightarrow \{0,1\}^m$ is a preimage resistant bijection. Define $h : \{0,1\}^{2m} \rightarrow \{0,1\}^m$ as follows. Given $x \in \{0,1\}^{2m}$, write

$$x = x' \parallel x''$$

where $x', x'' \in \{0,1\}^m$. Then define

$$h(x) = f(x' \oplus x'').$$

Prove that h is not second preimage resistant.

Answer: We are given $x = x' \parallel x''$. Let $x_0 \in \{0,1\}^m$, $x_0 \neq 0\,0 \cdots 0$. Define $x_1' = x' \oplus x_0$, $x_1'' = x'' \oplus x_0$ and $x_1 = x_1' \parallel x_1''$. Then $x \neq x_1$ and $h(x) = h(x_1)$.

4.7 For $M = 365$ and $15 \leq q \leq 30$, compare the exact value of ϵ given by the formula in the statement of Theorem 4.4 with the estimate for ϵ derived in the proof of that theorem.

Answer: Note: the estimate is derived after the proof of Theorem 4.4.

Define ϵ_1 to denote the exact probability, as computed in Theorem 4.4; and define $\epsilon_2 = 1 - e^{-q(q-1)/(2M)}$. Values of ϵ_1 and ϵ_2 are tabulated as follows:

q	ϵ_1	ϵ_2
15	.2529013198	.2499918703
16	.2836040053	.2801893756
17	.3150076653	.3110611335
18	.3469114179	.3424129197
19	.3791185260	.3740552376
20	.4114383836	.4058051275
21	.4436883352	.4374878054
22	.4756953077	.4689381108
23	.5072972343	.5000017522
24	.5383442579	.5305363394
25	.5686997040	.5604121995
26	.5982408201	.5895129752
27	.6268592823	.6177360099
28	.6544614723	.6449925266
29	.6809685375	.6712076120
30	.7063162427	.6963200177

4.8 Suppose $h : \mathcal{X} \rightarrow \mathcal{Y}$ is a hash function where $|\mathcal{X}|$ and $|\mathcal{Y}|$ are finite and $|\mathcal{X}| \geq 2|\mathcal{Y}|$. Suppose that H is balanced (i.e.,

$$|h^{-1}(y)| = \frac{|\mathcal{X}|}{|\mathcal{Y}|}$$

for all $y \in \mathcal{Y}$). Finally, suppose OraclePreimage is an (ϵ, q)-algorithm for Preimage, for the fixed hash function h. Prove that CollisionToPreimage is an $(\epsilon/2, q+1)$-algorithm for Collision, for the fixed hash function h.

Answer: We compute as follows:

$$\begin{aligned}
&\mathbf{Pr}[\textsc{CollisionToPreimage} \text{ succeeds}] \\
&= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{Pr}[\textsc{CollisionToPreimage} \text{ succeeds}|x] \\
&= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left(\mathbf{Pr}[\textsc{OraclePreimage} \text{ succeeds}|h(x)] \times \frac{\frac{|\mathcal{X}|}{|\mathcal{Y}|} - 1}{\frac{|\mathcal{X}|}{|\mathcal{Y}|}} \right) \\
&= \frac{1}{|\mathcal{X}|} \left(1 - \frac{|\mathcal{Y}|}{|\mathcal{X}|}\right) \sum_{x \in \mathcal{X}} \mathbf{Pr}[\textsc{OraclePreimage} \text{ succeeds}|h(x)] \\
&= \frac{1}{|\mathcal{X}|} \left(1 - \frac{|\mathcal{Y}|}{|\mathcal{X}|}\right) \frac{|\mathcal{X}|}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \mathbf{Pr}[\textsc{OraclePreimage} \text{ succeeds}|y] \\
&= \frac{1}{|\mathcal{Y}|} \left(1 - \frac{|\mathcal{Y}|}{|\mathcal{X}|}\right) \sum_{y \in \mathcal{Y}} \mathbf{Pr}[\textsc{OraclePreimage} \text{ succeeds}|y] \\
&\geq \frac{1}{|\mathcal{Y}|} \left(1 - \frac{|\mathcal{Y}|}{|\mathcal{X}|}\right) |\mathcal{Y}|\, \epsilon \\
&\geq \frac{\epsilon}{2}.
\end{aligned}$$

4.9 Suppose $h_1 : \{0,1\}^{2m} \rightarrow \{0,1\}^m$ is a collision resistant hash function.

(a) Define $h_2 : \{0,1\}^{4m} \rightarrow \{0,1\}^m$ as follows:

1. Write $x \in \{0,1\}^{4m}$ as $x = x_1 \parallel x_2$, where $x_1, x_2 \in \{0,1\}^{2m}$.
2. Define $h_2(x) = h_1(h_1(x_1) \parallel h_1(x_2))$.

Prove that h_2 is collision resistant.

Answer: Suppose that we have found a collision for h_2, say $h_2(x) = h_2(x')$ where $x \neq x'$. Denote $x = x_1 \parallel x_2$ and $x' = x'_1 \parallel x'_2$.

First, suppose that $h_1(x_1) \neq h_1(x'_1)$. Then

$$h_1(x_1) \parallel h_1(x_2) \neq h_1(x'_1) \parallel h_1(x'_2)$$

and

$$h_1(h_1(x_1) \parallel h_1(x_2)) = h_1(h_1(x'_1) \parallel h_1(x'_2)).$$

Therefore we have found a collision for h_1.

If $h_1(x_2) \neq h_1(x'_2)$, then we have a collision for h_1 by a similar argument.

Therefore we can assume that $h_1(x_1) = h_1(x'_1)$ and $h_1(x_2) = h_1(x'_2)$. Because $x \neq x'$, it follows that $(x_1, x_2) \neq (x'_1, x'_2)$. Therefore $x_1 \neq x'_1$ or $x_2 \neq x'_2$. In either of these two cases, we have a collision for h_1.

We conclude that we can always find a collision for h_1, given a collision for h_2.

(b) For an integer $i \geq 2$, define a hash function $h_i : \{0,1\}^{2^i m} \to \{0,1\}^m$ recursively from h_{i-1}, as follows:

1. Write $x \in \{0,1\}^{2^i m}$ as $x = x_1 \parallel x_2$, where $x_1, x_2 \in \{0,1\}^{2^{i-1} m}$.
2. Define $h_i(x) = h_1(h_{i-1}(x_1) \parallel h_{i-1}(x_2))$.

Prove that h_i is collision resistant.

Answer: Suppose that we have found a collision for h_i, say $h_i(x) = h_i(x')$ where $x \neq x'$. Denote $x = x_1 \parallel x_2$ and $x' = x_1' \parallel x_2'$.

First, suppose that $h_{i-1}(x_1) \neq h_{i-1}(x_1')$. Then

$$h_{i-1}(x_1) \parallel h_{i-1}(x_2) \neq h_{i-1}(x_1') \parallel h_{i-1}(x_2')$$

and

$$h_1(h_{i-1}(x_1) \parallel h_{i-1}(x_2)) = h_1(h_{i-1}(x_1') \parallel h_{i-1}(x_2')).$$

Therefore we have found a collision for h_1.

If $h_{i-1}(x_2) \neq h_{i-1}(x_2')$, then we have a collision for h_1 by a similar argument.

Therefore we can assume that $h_{i-1}(x_1) = h_{i-1}(x_1')$ and $h_{i-1}(x_2) = h_{i-1}(x_2')$. Because $x \neq x'$, it follows that $(x_1, x_2) \neq (x_1', x_2')$. Therefore $x_1 \neq x_1'$ or $x_2 \neq x_2'$. In either of these two cases, we have a collision for h_{i-1}.

We conclude that we can always find a collision for at least one of h_1 or h_{i-1}, given a collision for h_i.

4.10 In this exercise, we consider a simplified version of the Merkle-Damgård construction. Suppose

$$\mathsf{compress} : \{0,1\}^{m+t} \to \{0,1\}^m,$$

where $t \geq 1$, and suppose that

$$x = x_1 \parallel x_2 \parallel \cdots \parallel x_k,$$

where

$$|x_1| = |x_2| = \cdots = |x_k| = t.$$

We study the following iterated hash function:

```
Algorithm 4.1: SIMPLIFIED MERKLE-DAMGÅRD($x, k, t$)

external compress
$z_1 \leftarrow 0^m \parallel x_1$
$g_1 \leftarrow$ compress($z_1$)
for $i \leftarrow 1$ to $k - 1$
  do { $z_{i+1} \leftarrow g_i \parallel x_{i+1}$
     { $g_{i+1} \leftarrow$ compress($z_{i+1}$)
$h(x) \leftarrow g_{k+1}$
return ($h(x)$)
```

Suppose that compress is collision resistant, and suppose further that compress is zero preimage resistant, which means that it is hard to find $z \in \{0,1\}^{m+t}$ such that $\mathsf{compress}(z) = 0^m$. Under these assumptions, prove that h is collision resistant.

Answer: Note: In the seond last line of Algorithm 4.9, "g_{k+1}" should be replaced by "g_k".

Suppose that $h(x) = h(x')$ where $x \neq x'$. We consider two cases:

(a) $|x| = |x'| = kt$ for some positive integer k, and

(b) $|x| = kt$ and $|x'| = \ell t$, where k and ℓ are positive integers such that $\ell > k$.

We consider the two cases in turn.

(a) We have $g_k = g'_k$. If $z_k \neq z'_k$, then we have a collision for compress and we're done, so we assume that $z_k = z'_k$. This implies that $g_{k-1} = g'_{k-1}$ and $x_k = x'_k$.

Now if $z_{k-1} \neq z'_{k-1}$, then we have a collision, so we assume $z_{k-1} = z'_{k-1}$, which implies that $g_{k-2} = g'_{k-2}$ and $x_{k-1} = x'_{k-1}$.

Continuing to work backwards, either we find a collision for compress, or we have $x_i = x'_i$ for $i = k, k-1, \ldots, 1$. But then $x = x'$, a contradiction. We conclude that we always find a collision for compress in this case.

(b) We have $g_k = g'_\ell$. If $z_k \neq z'_\ell$, then we have a collision for compress and we're done, so we assume that $z_k = z'_\ell$. This implies that $g_{k-1} = g'_{\ell-1}$ and $x_k = x'_\ell$.

Now if $z_{k-1} \neq z'_{\ell-1}$, then we have a collision, so we assume $z_{k-1} = z'_{\ell-1}$, which implies that $g_{k-2} = g'_{\ell-2}$ and $x_{k-1} = x'_{\ell-1}$.

Continuing to work backwards, either we find a collision for compress, or we eventually reach the situation where $z_1 = z'_{\ell-k+1}$. Then $0^m = g'_{\ell-k} = \text{compress}(z'_{\ell-k})$, so compress is not zero preimage resistant. Therefore we either find a collision or a zero preimage for compress in this case.

4.11 A message authentication code can be produced by using a block cipher in CFB mode instead of CBC mode. Given a sequence of plaintext blocks, $x_1 \cdots x_n$, suppose we define the initialization vector IV to be x_1. Then encrypt the sequence $x_2 \cdots x_n$ using key K in CFB mode, obtaining the ciphertext sequence $y_1 \cdots y_{n-1}$ (note that there are only $n-1$ ciphertext blocks). Finally, define the MAC to be $e_K(y_{n-1})$. Prove that this MAC is identical to the MAC produced in Section 3.7 using CBC mode.

Answer: Using CFB mode, we obtain the following:

$$\begin{aligned}
\text{IV} &= x_1 \\
y_1 &= e_K(x_1) \oplus x_2 \\
y_2 &= e_K(y_1) \oplus x_3 \\
y_3 &= e_K(y_2) \oplus x_4 \\
\vdots\ &\vdots\ \vdots \\
y_{n-1} &= e_K(y_{n-2}) \oplus x_n \\
\text{MAC} &= e_K(y_{n-1}).
\end{aligned}$$

Using CBC mode with $\mathrm{IV} = 0\,0 \cdots 0$, we obtain the following:

$$\begin{aligned} \mathrm{IV} &= 0\,0 \cdots 0 \\ y'_1 &= e_K(x_1) \\ y'_2 &= e_K(y'_1 \oplus x_2) \\ y'_3 &= e_K(y'_2 \oplus x_3) \\ \vdots & \vdots \vdots \\ y'_n &= e_K(y'_{n-1} \oplus x_n) \\ \mathrm{MAC}' &= y'_n. \end{aligned}$$

It is easy to prove by induction on i that $y_i = y'_i \oplus x_{i+1}$, $1 \leq i \leq n-1$. Finally, we have

$$\begin{aligned} \mathrm{MAC} &= e_K(y_{n-1}) \\ &= e_K(y'_{n-1} \oplus x_n) \\ &= y'_n \\ &= \mathrm{MAC}'. \end{aligned}$$

Therefore the same MAC is produced by both methods.

4.12 Suppose that $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is an endomorphic cryptosystem with $\mathcal{P} = \mathcal{C} = \{0,1\}^m$. Let $n \geq$ be an 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:

$$h_K(y_1, \ldots y_n) = e_K(y_1) \oplus \cdots \oplus e_K(y_n).$$

Prove that $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$ is not a secure message authentication code as follows.

Note: you should assume $n \geq 2$ in this question.

(a) Prove the existence of a $(1,1)$-forger for this hash family.

Answer: Let $y_1, \ldots, y_n \in \{0,1\}^m$, $y_1 \neq y_2$. Request the MAC of $(y_1, \ldots, y_n)$, say y_0. Then y_0 is a forged MAC for the new message $(y_2, y_1, y_3, \ldots, y_n)$.

(b) Prove the existence of a $(1,2)$-forger for this hash family which can forge the MAC for an arbitrary message $(y_1, \ldots, y_n)$ (this is called a *selective forgery*; the forgeries previously considered are examples of *existential forgeries*). Note that the difficult case is when $y_1 = \cdots = y_n$.

Answer: Note: We actually construct a $(1,1)$-forger.

First, suppose that $y_i \neq y_j$ for some i, j. Define

$$y'_k = \begin{cases} y_k & \text{if } k \neq i, j \\ y_i & \text{if } k = j \\ y_j & \text{if } k = i. \end{cases}$$

Request the MAC of $(y'_1, \ldots, y'_n)$, say y_0. Then y_0 is a forged MAC for the message $(y_1, \ldots, y_n)$.

Now, suppose that $y_1 = \cdots = y_n$. If n is even, then $h_K(y_1, \ldots, y_1) = 0$ (i.e., we have a $(1,0)$-forgery). If n is odd, then let $y' \neq y_1$ and request the MAC of $(y', \ldots, y', y_1)$, say y_0. Then y_0 is a forged MAC for the message $(y_1, \ldots, y_1)$.

4.13 Suppose that $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is an endomorphic cryptosystem with $\mathcal{P} = \mathcal{C} = \{0,1\}^m$. Let $n \geq$ be an 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:

$$h_K(y_1, \dots, y_n) = e_K(y_1) + 3e_K(y_2) + \cdots + (2n-1)e_K(y_n) \bmod 2^m.$$

Note: you should assume $n \geq 2$ in this question.

(a) When n is odd, prove the existence of a $(1,2)$-forger for this hash family.

Answer: Suppose we request the MAC of $(x, x, \dots, x)$, say x'. Then $n^2 e_K(x) \equiv x' \pmod{2^m}$. Since n is odd, it follows that the inverse of n^2 exists modulo 2^m, which we denote by $n^{-2} \bmod 2^m$. Then $e_K(x) \equiv x' n^{-2} \pmod{2^m}$, so $e_K(x)$ can be computed, given x'.

Next, we request the MAC of $(y, y, \dots, y)$, say y', where $y \neq x$, and solve for $e_K(y)$. Now we can compute the MAC of $(x, y, \dots, y)$, for example, to be $(n^2 - 1)e_K(y) + e_K(x) \bmod 2^m$. This is a valid, forged MAC.

(b) When $n = 2$, prove the existence of a $(1/8, 2)$-forger for this hash family, as follows:

1. Request the MACs of (x, y) and (y, x). Suppose that $a = h_K(x, y)$ and $b = h_K(y, x)$.

2. Show that there are exactly eight ordered pairs (x', y') such that $x' = e_K(x)$, $y' = e_K(y)$ is consistent with the given MAC values a and b.

3. Choose one of these eight values for x' at random, and output the possible forgery $(x, x), x'$. Prove that this is a valid forgery with probability $1/8$.

Answer: Note: You should assume here that $m \geq 3$ and $y \neq x$. Also, the forgery to be outputted should be computed as $4x' \bmod 2^m$.

The system of two congruences $x' + 3y' \equiv a \pmod{2^m}$, $y' + 3x' \equiv b \pmod{2^m}$ has at least one solution. Writing $y' = b - 3x' \bmod 2^m$ and substituting into the other congruence, we obtain $x' + 3b - 9y' \equiv a \pmod{2^m}$, or $8y' \equiv 3b - a \pmod{2^m}$. This has at least one solution, so $3b - a \equiv 0 \pmod{8}$. Then it can be shown that this congruence has exactly eight solutions modulo 2^m, namely $y' = (3b-a)/8 + i2^{m-3} \bmod 2^m, 0 \leq i \leq 7$. For each y', the value of x' is defined uniquely, via the congruence $x' + 3y' \equiv a \pmod{2^m}$. Therefore there are exactly eight solutions for the pair (x', y').

Now choose one of the eight possible values of x'. Define $4x' \bmod 2^m$ to be the forged MAC for the message (x, x). This MAC will be valid if $x' = e_K(x)$, which is true with probability $1/8$.

(c) Prove the existence of a $(1,3)$-forger for this hash family which can forge the MAC for an arbitrary message $(y_1, \dots, y_n)$.

Answer: Choose $x \neq y_1, \dots, y_n$. Request the following three MACs:

i. z_1, the MAC of $(y_1, x, \dots, x)$;

ii. z_2, the MAC of $(x, y_2, \dots, y_n)$; and

iii. z_3, the MAC of $(x, x, \dots, x)$.

Then it is easy to see that the MAC of $(y_1, y_2, \dots, y_n)$ is $z_1 + z_2 - z_3 \bmod 2^m$.

4.14 Suppose that $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$ is a strongly universal (N, M)-hash family.

(a) If $|\mathcal{K}| = M^2$, show that there exists a $(1,2)$-forger for this hash family (i.e., $Pd_2 = 1$).

Answer: Choose any $x, x' \in \mathcal{X}$ such that $x \neq x'$. Request the MACs of x

and x', which we denote y, y', respectively. There is a unique key K such that $h_K(x) = y$ and $h_K(x') = y'$. Now given any $x'' \neq x, x'$, it is possible to compute the forged MAC $h_K(x'')$ because the key K is known.

(b) (This generalizes the result proven in part (a).) Denote $\lambda = |\mathcal{K}|/M^2$. Prove there exists a $(1/\lambda, 2)$-forger for this hash family (i.e., $Pd_2 \geq 1/\lambda$).
Answer: Choose any $x, x' \in \mathcal{X}$ such that $x \neq x'$. Request the MACs of x and x', which we denote y, y', respectively. There are exactly λ keys, say $K_1, \dots, K_\lambda$ such that $h_{K_i}(x) = y$ and $h_{K_i}(x') = y'$ for $1 \leq i \leq \lambda$. Choose $K \in \{K_1, \dots, K_\lambda\}$ randomly. Now given any $x'' \neq x, x'$, the MAC $h_K(x'')$ is valid with probability at least $1/\lambda$, because the probability that K is the correct key is $1/\lambda$.

4.15 Compute Pd_0 and Pd_1 for the following authentication code, represented in matrix form:

key	1	2	3	4
1	1	1	2	3
2	1	2	3	1
3	2	1	3	1
4	2	3	1	2
5	3	2	1	3
6	3	3	2	1

Answer: $Pd_0 = 1/2$; the pair $(4, 1)$ will be valid with this probability.

Define a_{ij} to denote the probability of forging a MAC for a new message, given that the MAC of i is j (where $1 \leq i \leq 4$, $1 \leq j \leq 3$). It is easy to verify the following:

i	j	a_{ij}	optimal forgery
1	1	$1/2$	$(2,1)$
1	2	$1/2$	$(2,1)$
1	3	$1/2$	$(2,2)$
2	1	$1/2$	$(1,1)$
2	2	$1/2$	$(1,1)$
2	3	$1/2$	$(1,2)$
3	1	$1/2$	$(1,2)$
3	2	$1/2$	$(1,1)$
3	3	1	$(4,1)$
4	1	$2/3$	$(3,3)$
4	2	1	$(1,2)$
4	3	$1/2$	$(1,1)$

Then

$$Pd_1 = \max\{\min\{a_{ij} : 1 \leq j \leq 3\} : 1 \leq i \leq 4\} = \frac{1}{2}.$$

4.16 Let p be an odd prime. For $a, b \in \mathbb{Z}_p$, define $f_{(a,b)} : \mathbb{Z}_p \rightarrow \mathbb{Z}_p$ by the rule

$$f_{(a,b)}(x) = (x + a)^2 + b \bmod p.$$

Prove that $(\mathbb{Z}_p, \mathbb{Z}_p, \mathbb{Z}_p \times \mathbb{Z}_p, \{f_{(a,b)} : a, b \in \mathbb{Z}_p\})$ is a strongly universal (p, p)-hash family.

Answer: Suppose that $x, x', y, y' \in \mathbb{Z}_p$, where $x \neq x'$. We will show that there is a unique key $(a, b) \in \mathbb{Z}_p \times \mathbb{Z}_p$ such that $(x + a)^2 + b \equiv y \pmod{p}$ and

$(x'+a)^2+b \equiv y' \pmod{p}$. Subtracting these two equations, we have

$$\begin{aligned}(x+a)^2-(x'+a)^2 &\equiv y-y' \pmod{p}\\ x^2-(x')^2+2a(x-x') &\equiv y-y' \pmod{p}\\ x+x'+2a &\equiv (y-y')(x-x')^{-1} \pmod{p}\\ a &= 2^{-1}((y-y')(x-x')^{-1}-(x+x')) \bmod p.\end{aligned}$$

Now that a has been determined uniquely (modulo p), we can solve for b, because $b = y-(x+a)^2 \bmod p$.

4.17 Let $k \geq 1$ be an integer. An (N, M) hash family, $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, is *strongly k-universal* provided that the following condition is satisfied for all choices of k distinct elements $x_1, x_2, \ldots, x_k \in \mathcal{X}$ and for all choices of k (not necessarily distinct) elements $y_1, \ldots, y_k \in \mathcal{Y}$:

$$|\{K \in \mathcal{K} : h_K(x_i) = y_1 \text{ for } 1 \leq i \leq k\}| = \frac{|\mathcal{K}|}{M^k}.$$

(a) Prove that a strongly k-universal hash family is strongly ℓ-universal for all ℓ such that $1 \leq \ell \leq k$.

Answer: Note: in the definition of strongly k-universal, "$h_K(x_i) = y_1$" should be replaced by "$h_K(x_i) = y_i$".

Without loss of generality, suppose that $\ell < k$. Suppose that $x_1, x_2, \ldots, x_\ell \in \mathcal{X}$ are distinct, and suppose that $y_1, \ldots, y_\ell \in \mathcal{Y}$. Let $x_{\ell+1}, \ldots, x_k \in \mathcal{X}$ be chosen such that $x_1, x_2, \ldots, x_k$ are all distinct. Now, for any $(k-\ell)$-tuple $(y_{\ell+1}, \ldots, y_k) \in \mathcal{Y}^{k-\ell}$, it holds that

$$|\{K \in \mathcal{K} : h_K(x_i) = y_i \text{ for } 1 \leq i \leq k\}| = \frac{|\mathcal{K}|}{M^k}.$$

Then it is clear that

$$\begin{aligned}&|\{K \in \mathcal{K} : h_K(x_i) = y_i \text{ for } 1 \leq i \leq \ell\}|\\ &= \sum_{(y_{\ell+1},\ldots,y_k)\in\mathcal{Y}^{k-\ell}} |\{K \in \mathcal{K} : h_K(x_i) = y_i \text{ for } 1 \leq i \leq k\}|\\ &= M^{k-\ell} \times \frac{|\mathcal{K}|}{M^k}\\ &= \frac{|\mathcal{K}|}{M^\ell},\end{aligned}$$

as desired.

(b) Let p be prime and let $k \geq 1$ be an integer. For all k-tuples $(a_0, \ldots, a_{k-1}) \in (\mathbb{Z}_p)^k$, define $f_{(a_0,\ldots,a_{k-1})} : \mathbb{Z}_p \to \mathbb{Z}_p$ by the rule

$$f_{(a_0,\ldots,a_{k-1})}(x) = \sum_{i=0}^{k-1} a_i x^i \bmod p.$$

Prove that $\left(\mathbb{Z}_p, \mathbb{Z}_p, (\mathbb{Z}_p)^k, \{f_{(a_0,\ldots,a_{k-1})} : (a_0, \ldots, a_{k-1}) \in (\mathbb{Z}_p)^k\}\right)$ is a strongly k-universal (p, p) hash family.

HINT Use the fact that any degree d polynomial over a field has at most d roots.

Answer: Let $x_1, x_2, \ldots, x_k \in \mathcal{X}$ be k distinct elements. There are p^k possible keys, and p^k possible k-tuples $(y_1, \ldots, y_k) \in (\mathbb{Z}_p)^k$. We will show that, given any k-tuple $(y_1, \ldots, y_k) \in (\mathbb{Z}_p)^k$, there is exactly one key $(a_0, \ldots, a_{k-1}) \in (\mathbb{Z}_p)^k$ such that $f_{(a_0,\ldots,a_{k-1})}(x_i) = y_i$ for $1 \leq i \leq k$. Suppose this is not the case. Then there must exist two different keys $(a_0, \ldots, a_{k-1}) \neq (a'_0, \ldots, a'_{k-1})$ such that $f_{(a_0,\ldots,a_{k-1})}(x_i) = f_{(a'_0,\ldots,a'_{k-1})}(x_i) = y_i$ for $1 \leq i \leq k$. This implies that

$$\sum_{i=0}^{k-1}(a_i - a'_i)x^i \equiv 0 \pmod{p}$$

has at least k solutions in $\mathbb{Z}_p$, namely $x_1, x_2, \ldots, x_k$. In other words, the polynomial

$$g(x) = \sum_{i=0}^{k-1}(a_i - a'_i)x^i$$

has at least k distinct roots in the field $\mathbb{Z}_p$. The two k-tuples $(a_0, \ldots, a_{k-1})$ and $(a'_0, \ldots, a'_{k-1})$ are different, so the polynomial $g(x)$ is not the zero polynomial. But a non-zero polynomial of degree at most $k - 1$ cannot have k distinct roots in a field, so we have a contradiction. This contradiction establishes the desired result.

5

The RSA Cryptosystem and Factoring Integers

Exercises

5.1 In Algorithm 5.1, prove that

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots = \gcd(r_{m-1}, r_m) = r_m$$

and, hence, $r_m = \gcd(a, b)$.

Answer: Suppose that $0 \leq i \leq m-2$. Then have that $r_i = q_{i+1} r_{i+1} + r_{i+2}$. If $d | r_i$ and $d | r_{i+1}$, then $d | r_{i+2}$. Also, if $d | r_{r+1}$ and $d | r_{i+2}$, then $d | r_i$. This proves that

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots = \gcd(r_{m-1}, r_m).$$

Now, using the equation $r_{m-1} = q_m r_m$, we have that $\gcd(r_{m-1}, r_m) = r_m$, and the result is proven.

5.2 Suppose that $a > b$ in Algorithm 5.1.

(a) Prove that $r_i \geq 2 r_{i+2}$ for all i such that $0 \leq i \leq m-2$.

Answer: $r_i = q_{i+1} r_{i+1} + r_{i+2} \geq r_{i+1} + r_{i+2} > 2 r_{i+2}$ for $0 \leq i \leq m-2$.

(b) Prove that m is $O((\log a)^2)$.

Answer: Suppose first that m is even. Then $a = r_0 > 2r_2 > \cdots > 2^{m/2} r_m \geq 2^{m/2}$. Therefore $m < 2 \log_2 a$. If m is odd, then it can be shown in a similar fashion that $m < 2 \log_2 a + 1$. In either case, m is $O(\log a)$.

(c) Prove that m is $O((\log b)^2)$.

Answer: Suppose first that m is odd. Then $b = r_1 > 2r_3 > \cdots > 2^{(m-1)/2} r_m \geq 2^{(m-1)/2}$. Therefore $m < 2 \log_2 b + 1$. If m is even, then it can be shown in a similar fashion that $m < 2 \log_2 a + 2$. In either case, m is $O(\log b)$.

5.3 Use the EXTENDED EUCLIDEAN ALGORITHM to compute the following multiplicative inverses:

(a) $17^{-1} \bmod 101$

Answer: $17^{-1} \bmod 101 = 6$.

(b) $357^{-1} \bmod 1234$

Answer: $357^{-1} \bmod 1234 = 1075$.

(c) $3125^{-1} \bmod 9987$.

Answer: $3125^{-1} \bmod 9987 = 1844$.

5.4 Compute $\gcd(57, 93)$, and find integers s and t such that $57s + 93t = \gcd(57, 93)$.
Answer: $\gcd(57, 93) = 3 = 18 \times 57 - 11 \times 93$.

5.5 Suppose $\chi : \mathbb{Z}_{105} \to \mathbb{Z}_3 \times \mathbb{Z}_5 \times \mathbb{Z}_7$ is defined as

$$\chi(x) = (x \bmod 3, x \bmod 5, x \bmod 7).$$

Give an explicit formula for the function χ^{-1}, and use it to compute $\chi^{-1}(2, 2, 3)$.
Answer: $\chi^{-1}(a_1, a_2, a_3) = 70a_1 + 21a_2 + 15a_3 \bmod 105$, and $\chi^{-1}(2, 2, 3) = 17$.

5.6 Solve the following system of congruences:

$$x \equiv 12 \pmod{25}$$

$$x \equiv 9 \pmod{26}$$

$$x \equiv 23 \pmod{27}.$$

Answer: $x = 14387$.

5.7 Solve the following system of congruences:

$$13x \equiv 4 \pmod{99}$$

$$15x \equiv 56 \pmod{101}.$$

HINT First use the EXTENDED EUCLIDEAN ALGORITHM, and then apply the Chinese remainder theorem.

Answer: $x = 7471$.

5.8 Use Theorem 5.8 to find the smallest primitive element modulo 97.
Answer: $2^{48} \bmod 97 = 1$, $3^{48} \bmod 97 = 1$, $4^{48} \bmod 97 = 1$, $5^{48} \bmod 97 = 96$ and $5^{32} \bmod 97 = 35$. Therefore the smallest primitive root modulo 97 is 5.

5.9 Suppose that $p = 2q + 1$, where p and q are odd primes. Suppose further that $\alpha \in {\mathbb{Z}_p}^*$, $\alpha \not\equiv \pm 1 \pmod{p}$. Prove that α is a primitive element modulo p if and only if $\alpha^q \equiv -1 \pmod{p}$.
Answer: This follows immediately from Theorem 5.8, which (in this case) states that α is a primitive element modulo p if and only if $\alpha^q \not\equiv -1 \pmod{p}$ and $\alpha^2 \not\equiv 1 \pmod{p}$. But $\alpha^2 \equiv 1 \pmod{p}$ if and only if $\alpha \equiv \pm 1 \pmod{p}$. We have assumed that $\alpha \not\equiv \pm 1 \pmod{p}$, so the result follows.

5.10 Suppose that $n = pq$, where p and q are distinct odd primes and $ab \equiv 1 \pmod{(p-1)(q-1)}$. The RSA encryption operation is $e(x) = x^b \bmod n$ and the decryption operation is $d(y) = y^a \bmod n$. We proved that $d(e(x)) = x$ if $x \in {\mathbb{Z}_n}^*$. Prove that the same statement is true for any $x \in \mathbb{Z}_n$.

HINT Use the fact that $x_1 \equiv x_2 \pmod{pq}$ if and only if $x_1 \equiv x_2 \pmod{p}$ and $x_1 \equiv x_2 \pmod{q}$. This follows from the Chinese remainder theorem.

Answer: Suppose $x \not\equiv 0 \pmod{p}$. Then, for some integer $k > 0$, it holds that

$$x^{ab} = x^{1+k(p-1)(q-1)} \equiv x \times x^{k(p-1)(q-1)} \equiv x \pmod{p}.$$

If $x \equiv 0 \pmod{p}$, then $x^{ab} \equiv x \equiv 0 \pmod{p}$. Therefore $x^{ab} \equiv x \pmod{p}$ for any $x \in \mathbb{Z}_p$. Similarly, $x^{ab} \equiv x \pmod{q}$ for any $x \in \mathbb{Z}_q$. Now, applying the hint, $x^{ab} \equiv x \pmod{n}$ for any $x \in \mathbb{Z}_n$.

5.11 For $n = pq$, where p and q are distinct odd primes, define

$$\lambda(n) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}.$$

Suppose that we modify the *RSA Cryptosystem* by requiring that $ab \equiv 1 \pmod{\lambda(n)}$.

(a) Prove that encryption and decryption are still inverse operations in this modified cryptosystem.

Answer: Denote $d = \gcd(p-1, q-1)$, $p - 1 = p'd$ and $q - 1 = q'd$. Then

$$\lambda(n) = p'q'd = (p-1)q' = p'(q-1).$$

We have that $ab \equiv 1 \pmod{\lambda(n)}$, so

$$ab = k\lambda(n) + 1 = k(p-1)q' + 1$$

for some positive integer k. Then

$$x^{ab} \equiv x^{k(p-1)q'+1} \pmod{p} \equiv x \pmod{p}.$$

Similarly,

$$x^{ab} \equiv x^{k(q-1)p'+1} \pmod{q} \equiv x \pmod{q}.$$

Since $x^{ab} \equiv x \pmod{p}$ and $x^{ab} \equiv \pmod{q}$, it follows immediately that $x^{ab} \equiv x \pmod{n}$.

(b) If $p = 37$, $q = 79$, and $b = 7$, compute a in this modified cryptosystem, as well as in the original *RSA Cryptosystem*.

Answer: $d = 6$, $\lambda(n) = 468$ and $\phi(n) = 2808$. $b^{-1} \bmod \lambda(n) = 67$ and $b^{-1} \bmod \phi(n) = 2407$.

5.12 Two samples of RSA ciphertext are presented in Tables 5.1 and 5.2. Your task is to decrypt them. The public parameters of the system are $n = 18923$ and $b = 1261$ (for Table 5.1) and $n = 31313$ and $b = 4913$ (for Table 5.2). This can be accomplished as follows. First, factor n (which is easy because it is so small). Then compute the exponent a from $\phi(n)$, and, finally, decrypt the ciphertext. Use the SQUARE-AND-MULTIPLY ALGORITHM to exponentiate modulo n.

In order to translate the plaintext back into ordinary English text, you need to know how alphabetic characters are "encoded" as elements in $\mathbb{Z}_n$. Each element of $\mathbb{Z}_n$ represents three alphabetic characters as in the following examples:

$$\begin{array}{lcrcr} DOG & \rightarrow & 3 \times 26^2 + 14 \times 26 + 6 & = & 2398 \\ CAT & \rightarrow & 2 \times 26^2 + 0 \times 26 + 19 & = & 1371 \\ ZZZ & \rightarrow & 25 \times 26^2 + 25 \times 26 + 25 & = & 17575. \end{array}$$

You will have to invert this process as the final step in your program.

Answer: The first plaintext was encrypted using the values $n = 18923 = 127 \times 149$ and $b = 1261$. Hence, $\phi(n) = 126 \times 148 = 18648$ and $a = 1261^{-1} \bmod 18648 = 5797$.

The first plaintext was taken from "The Diary of Samuel Marchbanks," by Robertson Davies, 1947. The first ciphertext element, $y = 12423$, is decrypted to $x = 5438$. We convert this to three letters as follows:

$$5438 \bmod 26 = 4$$

$$(5438 - 4)/26 = 209$$

$$209 \bmod 26 = 1$$

$$(209 - 1)/26 = 8$$

$$8 \bmod 26 = 8.$$

Therefore, the triple $8, 1, 4$ corresponds to the three letters i, b, e.

The complete plaintext is as follows:

TABLE 5.1
RSA ciphertext

12423	11524	7243	7459	14303	6127	10964	16399
9792	13629	14407	18817	18830	13556	3159	16647
5300	13951	81	8986	8007	13167	10022	17213
2264	961	17459	4101	2999	14569	17183	15827
12693	9553	18194	3830	2664	13998	12501	18873
12161	13071	16900	7233	8270	17086	9792	14266
13236	5300	13951	8850	12129	6091	18110	3332
15061	12347	7817	7946	11675	13924	13892	18031
2620	6276	8500	201	8850	11178	16477	10161
3533	13842	7537	12259	18110	44	2364	15570
3460	9886	8687	4481	11231	7547	11383	17910
12867	13203	5102	4742	5053	15407	2976	9330
12192	56	2471	15334	841	13995	17592	13297
2430	9741	11675	424	6686	738	13874	8168
7913	6246	14301	1144	9056	15967	7328	13203
796	195	9872	16979	15404	14130	9105	2001
9792	14251	1498	11296	1105	4502	16979	1105
56	4118	11302	5988	3363	15827	6928	4191
4277	10617	874	13211	11821	3090	18110	44
2364	15570	3460	9886	9988	3798	1158	9872
16979	15404	6127	9872	3652	14838	7437	2540
1367	2512	14407	5053	1521	297	10935	17137
2186	9433	13293	7555	13618	13000	6490	5310
18676	4782	11374	446	4165	11634	3846	14611
2364	6789	11634	4493	4063	4576	17955	7965
11748	14616	11453	17666	925	56	4118	18031
9522	14838	7437	3880	11476	8305	5102	2999
18628	14326	9175	9061	650	18110	8720	15404
2951	722	15334	841	15610	2443	11056	2186

> I became involved in an argument about modern painting, a subject upon which I am spectacularly ill-informed. However, many of my friends can become heated and even violent on the subject, and I enjoy their wrangles in a modest way. I am an artist myself and I have some sympathy with the abstractionists, although I have gone beyond them in my own approach to art. I am a lumpist. Two or three decades ago it was quite fashionable to be a cubist and to draw everything in cubes. Then there was a revolt by the vorticists who drew everything in whirls. We now have the abstractionists who paint everything in a very abstracted manner, but my own small works done on my telephone pad are composed of carefully shaded, strangely shaped lumps with traces of cubism, vorticism, and abstractionism in them. For those who possess the seeing eye, as a lumpist, I stand alone.

The second plaintext was encrypted using the values $n = 31313 = 173 \times 181$ and $b = 4913$. Hence, $\phi(n) = 172 \times 180 = 30960$ and $a = 4913^{-1} \bmod 30960 =$

TABLE 5.2
RSA ciphertext

6340	8309	14010	8936	27358	25023	16481	25809
23614	7135	24996	30590	27570	26486	30388	9395
27584	14999	4517	12146	29421	26439	1606	17881
25774	7647	23901	7372	25774	18436	12056	13547
7908	8635	2149	1908	22076	7372	8686	1304
4082	11803	5314	107	7359	22470	7372	22827
15698	30317	4685	14696	30388	8671	29956	15705
1417	26905	25809	28347	26277	7897	20240	21519
12437	1108	27106	18743	24144	10685	25234	30155
23005	8267	9917	7994	9694	2149	10042	27705
15930	29748	8635	23645	11738	24591	20240	27212
27486	9741	2149	29329	2149	5501	14015	30155
18154	22319	27705	20321	23254	13624	3249	5443
2149	16975	16087	14600	27705	19386	7325	26277
19554	23614	7553	4734	8091	23973	14015	107
3183	17347	25234	4595	21498	6360	19837	8463
6000	31280	29413	2066	369	23204	8425	7792
25973	4477	30989					

6497.

The second plaintext was taken from “Lake Wobegon Days,” by Garrison Keillor, 1985. It is as follows:

> Lake Wobegon is mostly poor sandy soil, and every spring the earth heaves up a new crop of rocks. Piles of rocks ten feet high in the corners of fields, picked by generations of us, monuments to our industry. Our ancestors chose the place, tired from their long journey, sad for having left the motherland behind, and this place reminded them of there, so they settled here, forgetting that they had left there because the land wasn’t so good. So the new life turned out to be a lot like the old, except the winters are worse.

5.13 A common way to speed up RSA decryption incorporates the Chinese remainder theorem, as follows. Suppose that $d_K(y) = y^a \bmod n$ and $n = pq$. Define $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$; and let $M_p = q^{-1} \bmod p$ and $M_q = p^{-1} \bmod q$. Then consider the following algorithm:

```
Algorithm 5.15: CRT-OPTIMIZED RSA DECRYPTION($n, d_p, d_q, M_p, M_q, y$)

  $x_p \leftarrow y^{d_p} \bmod p$
  $x_q \leftarrow y^{d_q} \bmod q$
  $x \leftarrow M_p q x_p + M_q p x_q \bmod n$
  return $(x)$
```

Algorithm 5.15 replaces an exponentiation modulo n by modular exponentiations modulo p and q. If p and q are ℓ-bit integers and exponentiation modulo an ℓ-

bit integer takes time $c\ell^3$, then the time to perform the required exponentiation(s) is reduced from $c(2\ell)^3$ to $2c\ell^3$, a savings of 75%. The final step, involving the Chinese remainder theorem, requires time $O(\ell^2)$ if d_p, d_q, M_p and M_q have been pre-computed.

(a) Prove that the value x returned by Algorithm 5.15 is, in fact, $y^d \bmod n$.
Answer:

$$\begin{aligned} x &\equiv M_p q x_p \pmod p \\ &\equiv q^{-1} q x_p \pmod p \\ &\equiv x_p \pmod p \\ &\equiv y^{d_p} \pmod p \\ &\equiv y^d \pmod p, \end{aligned}$$

because $d_p \equiv d \pmod{p-1}$. Similarly, $x \equiv y^d \pmod q$. Therefore $x \equiv y^d \pmod n$.

(b) Given that $p = 1511$ and $q = 2003$, compute d_p, d_q, M_p and M_q.
Answer: Note: A value of d needs to be specified in order to compute d_p and d_q.

Suppose we take $d = 1234577$. Then $d_p = 907$, $d_q = 1345$, $M_p = 777$ and $M_q = 973$.

(c) Given the above values of p and q, decrypt the ciphertext $y = 152702$ using Algorithm 5.15.
Answer: Note: Again, d needs to be specified, as in part (b).

Using $d = 1234577$, we obtain $x_p = 242$, $x_q = 1087$ and $x = 1443247$.

5.14 Prove that the *RSA Cryptosystem* is insecure against a chosen ciphertext attack. In particular, given a ciphertext y, describe how to choose a ciphertext $\hat{y} \neq y$, such that knowledge of the plaintext $\hat{x} = d_K(\hat{y})$ allows $x = d_K(y)$ to be computed.

HINT Use the multiplicative property of the *RSA Cryptosystem*, i.e., that

$$e_K(x_1)e_K(x_2) \bmod n = e_K(x_1 x_2 \bmod n).$$

Answer: Choose a random x_0 and compute $y_0 = e_K(x_0)$. Define $\hat{y} = y_0 y \bmod n$, and obtain the decryption $\hat{x} = d_K(\hat{y})$. Then compute $x = \hat{x} {x_0}^{-1} \bmod n$.

5.15 This exercise exhibits what is called a *protocol failure*. It provides an example where ciphertext can be decrypted by an opponent, without determining the key, if a cryptosystem is used in a careless way. The moral is that it is not sufficient to use a “secure” cryptosystem in order to guarantee “secure” communication.

Suppose Bob has an *RSA Cryptosystem* with a large modulus n for which the factorization cannot be found in a reasonable amount of time. Suppose Alice sends a message to Bob by representing each alphabetic character as an integer between 0 and 25 (i.e., $A \leftrightarrow 0$, $B \leftrightarrow 1$, etc.), and then encrypting each residue modulo 26 as a separate plaintext character.

(a) Describe how Oscar can easily decrypt a message which is encrypted in this way.
Answer: Oscar can encrypt each of the 26 possible plaintexts, and record the values of the corresponding 26 ciphertexts in a table. Then any ciphertext string can be decrypted by referring to the precomputed table.

(b) Illustrate this attack by decrypting the following ciphertext (which was encrypted using an *RSA Cryptosystem* with $n = 18721$ and $b = 25$) without factoring the modulus:

$$365, 0, 4845, 14930, 2608, 2608, 0.$$

Answer: The plaintext is $vanilla$.

5.16 This exercise illustrates another example of a protocol failure (due to Simmons) involving the *RSA Cryptosystem*; it is called the "common modulus protocol failure." Suppose Bob has an *RSA Cryptosystem* with modulus n and encryption exponent b_1, and Charlie has an *RSA Cryptosystem* with (the same) modulus n and encryption exponent b_2. Suppose also that $\gcd(b_1, b_2) = 1$. Now, consider the situation that arises if Alice encrypts the same plaintext x to send to both Bob and Charlie. Thus, she computes $y_1 = x^{b_1} \bmod n$ and $y_2 = x^{b_2} \bmod n$, and then she sends y_1 to Bob and y_2 to Charlie. Suppose Oscar intercepts y_1 and y_2, and performs the computations indicated in Algorithm 5.16.

```
Algorithm 5.16: RSA COMMON MODULUS DECRYPTION($n, b_1, b_2, y_1, y_2$)

$c_1 \leftarrow {b_1}^{-1} \bmod b_2$
$c_2 \leftarrow (c_1 b_1 - 1)/b_2$
$x_1 \leftarrow {y_1}^{c_1}({y_2}^{c_2})^{-1} \bmod n$
return $(x_1)$
```

(a) Prove that the value x_1 computed in Algorithm 5.16 is in fact Alice's plaintext, x. Thus, Oscar can decrypt the message Alice sent, even though the cryptosystem may be "secure."
Answer: We use the fact that $b_2 c_2 = b_1 c_1 - 1$. Working in $\mathbb{Z}_n$, we have that

$${y_1}^{c_1}({y_2}^{c_2})^{-1} = x^{b_1 c_1}(x^{b_2 c_2})^{-1} = x^{b_1 c_1 - b_1 c_2} = x.$$

(b) Illustrate the attack by computing x by this method if $n = 18721$, $b_1 = 43$, $b_2 = 7717$, $y_1 = 12677$ and $y_2 = 14702$.
Answer: $c_1 = 2692$, $c_2 = 15$, and $x = 15001$.

5.17 We give yet another protocol failure involving the *RSA Cryptosystem*. Suppose that three users in a network, say Bob, Bart and Bert, all have public encryption exponents $b = 3$. Let their moduli be denoted by n_1, n_2, n_3, and assume that n_1, n_2 and n_3, are pairwise relatively prime. Now suppose Alice encrypts the same plaintext x to send to Bob, Bart and Bert. That is, Alice computes $y_i = x^3 \bmod n_i$, $1 \leq i \leq 3$. Describe how Oscar can compute x, given y_1, y_2 and y_3, without factoring any of the moduli.

Answer: Consider the following system of three congruences:

$$z \equiv y_1 \bmod n_1$$

$$z \equiv y_2 \bmod n_2$$

$$z \equiv y_3 \bmod n_3.$$

Using the Chinese remainder theorem, it is easy to find the unique solution z to this system such that $0 \leq z < n_1 n_2 n_3$. However ,the integer x^3 is a solution to the same system, and $0 \leq x^3 < n_1 n_2 n_3$. Since the system has a unique solution modulo $n_1 n_2 n_3$, it must be the case that $x^3 = z$. Therefore $x = z^{1/3}$.

5.18 A plaintext x is said to be *fixed* if $e_K(x) = x$. Show that, for the *RSA Cryptosystem*, the number of fixed plaintexts $x \in \mathbb{Z}_n{}^*$ is equal to

$$\gcd(b-1, p-1) \times \gcd(b-1, q-1).$$

HINT Consider the following system of two congruences:

$$e_K(x) \equiv x \pmod{p},$$

$$e_K(x) \equiv x \pmod{q}.$$

Answer: $e_K(x) = x$ if and only if

$$x^b \equiv x \pmod{p} \quad \text{and}$$

$$x^b \equiv x \pmod{q}.$$

First, we determine the number of solutions $x \in \mathbb{Z}_p{}^*$ to the congruence $x^b \equiv x \pmod{p}$. Let $\alpha \in \mathbb{Z}_p{}^*$ be a primitive element. Then any $x \in \mathbb{Z}_p{}^*$ can be written uniquely in the form $x = \alpha^i \bmod p$, where $0 \leq i \leq p-2$. Further, $x^b \equiv x \pmod{p}$ if and only if $ib \equiv i \pmod{p-1}$, or $i(b-1) \equiv 0 \pmod{p-1}$. This congruence has $\gcd(b-1, p-1)$ solutions, namely

$$i = \frac{j(p-1)}{\gcd(b-1, p-1)},$$

$0 \leq j < \gcd(b-1, p-1)$. Therefore $x^b \equiv x \pmod{p}$ has $\gcd(b-1, p-1)$ solutions $x \in \mathbb{Z}_p{}^*$.

Similarly, $x^b \equiv x \pmod{q}$ has $\gcd(b-1, q-1)$ solutions $x \in \mathbb{Z}_q{}^*$. Using the Chinese reaminder theorem, it is clear that the number of solutions $x \in \mathbb{Z}_n{}^*$ to the system

$$x^b \equiv x \pmod{p},$$

$$x^b \equiv x \pmod{q}.$$

is exactly $\gcd(b-1, p-1) \times \gcd(b-1, q-1)$.

5.19 Suppose $\mathbf{A}$ is a deterministic algorithm which is given as input an RSA modulus n, an encryption exponent b, and a ciphertext y. $\mathbf{A}$ will either decrypt y or return no answer. Supposing that there are $\epsilon(n-1)$ ciphertexts which $\mathbf{A}$ is able to decrypt, show how to use $\mathbf{A}$ as an oracle in a Las Vegas decryption algorithm having success probability ϵ.

Answer: Note: You should assume that $\epsilon(n-1)$ is the number of non-zero ciphertexts that $\mathbf{A}$ can successfully decrypt.

Suppose we are given n, b and a ciphertext $y \in \mathbb{Z}_n$. If $y = 0$, then its decryption is $x = 0$. If $\gcd(y, n) > 1$, then it is possible to factor n, in which case y can easily be decrypted. Therefore we suppose that $\gcd(y, n) = 1$.

Now, the algorithm $\mathbf{B}$ should choose a random $x_1 \in \mathbb{Z}_n$, $x_1 \neq 0$; compute $y_1 = x_1{}^b \bmod n$; and compute $y' = yy_1 \bmod n$. Then call the algorithm $\mathbf{A}$ with input n, b, y'. If $\mathbf{A}$ returns a decryption of y', say x', then $x = x'/x_1 \bmod n$.

We need to analyze the success probability of $\mathbf{B}$. If $\gcd(y, n) > 1$, then $\mathbf{B}$ has success probability equal to 1. If $\gcd(y, n) = 1$, then y' is a random non-zero element of $\mathbb{Z}_n$, so the success probability is $\epsilon(n-1)/(n-1) = \epsilon$. Therefore, for any input y, the success probability of $\mathbf{B}$ is greater than ϵ.

5.20 Write a program to evaluate Jacobi symbols using the four properties presented in Section 5.4. The program should not do any factoring, other than dividing out powers of two. Test your program by computing the following Jacobi symbols:

$$\left(\frac{610}{987}\right), \left(\frac{20964}{1987}\right), \left(\frac{1234567}{11111111}\right).$$

Answer: The three Jacobi symbols are -1, 1 and -1, respectively.

5.21 For $n = 837$, 851 and 1189, find the number of bases b such that n is an Euler pseudo-prime to the base b.

Answer: The number of bases b is 10, 2 and 8 respectively.

5.22 The purpose of this question is to prove that the error probability of the Solovay-Strassen primality test is at most $1/2$. Let $\mathbb{Z}_n{}^*$ denote the group of units modulo n. Define

$$G(n) = \left\{ a : a \in \mathbb{Z}_n{}^*, \left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod{n} \right\}.$$

(a) Prove that $G(n)$ is a subgroup of $\mathbb{Z}_n{}^*$. Hence, by Lagrange's theorem, if $G(n) \neq \mathbb{Z}_n{}^*$, then

$$|G(n)| \leq \frac{|\mathbb{Z}_n{}^*|}{2} \leq \frac{n-1}{2}.$$

Answer: Suppose that $a, b \in G(n)$. Then

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod{n}$$

and

$$\left(\frac{b}{n}\right) \equiv b^{(n-1)/2} \pmod{n}.$$

It follows from the multiplicative rule of Jacobi symbols (page 176, property 3) that

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right) \equiv a^{(n-1)/2} b^{(n-1)/2} \pmod{n} \equiv (ab)^{(n-1)/2} \pmod{n}.$$

Therefore $ab \in G(n)$. Since $G(n)$ is a subset of a multiplicative finite group that is closed under the operation of multiplication, it must be a subgroup.

(b) Suppose $n = p^k q$, where p and q are odd, p is prime, $k \geq 2$, and $\gcd(p, q) = 1$. Let $a = 1 + p^{k-1} q$. Prove that

$$\left(\frac{a}{n}\right) \not\equiv a^{(n-1)/2} \pmod{n}.$$

HINT Use the binomial theorem to compute $a^{(n-1)/2}$.

Answer: We have that

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)^k \left(\frac{a}{q}\right) = 1.$$

On the other hand,

$$\begin{aligned} a^{(n-1)/2} &= \sum_{i=0}^{(n-1)/2} \binom{(n-1)/2}{i} (p^{k-1} q)^i \\ &\equiv 1 + \frac{n-1}{2} p^{k-1} q \pmod{n}. \end{aligned}$$

Suppose that $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod{n}$. Then

$$\frac{n-1}{2} p^{k-1} q \equiv 0 \pmod{n}.$$

This implies that

$$p^k q | \frac{n-1}{2} p^{k-1} q,$$

$$p \Big| \frac{n-1}{2},$$

and hence $n \equiv 1 \pmod{p}$. But $n \equiv 0 \pmod{p}$, so we have a contradiction.

(c) Suppose $n = p_1 \dots p_s$, where the p_i's are distinct odd primes. Suppose $a \equiv u \pmod{p_1}$ and $a \equiv 1 \pmod{p_2 p_3 \dots p_s}$, where u is a quadratic non-residue modulo p_1 (note that such an a exists by the Chinese remainder theorem). Prove that

$$\left(\frac{a}{n}\right) \equiv -1 \pmod{n},$$

but

$$a^{(n-1)/2} \equiv 1 \pmod{p_2 p_3 \dots p_s},$$

so

$$a^{(n-1)/2} \not\equiv -1 \pmod{n}.$$

Answer: On one hand, we have

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2 p_3 \dots p_s}\right) = (-1)(1) = -1.$$

If $a^{(n-1)/2} \equiv -1 \pmod{n}$, then $a^{(n-1)/2} \equiv -1 \pmod{p_2 p_3 \dots p_s}$. But $a^{(n-1)/2} \equiv 1 \pmod{p_2 p_3 \dots p_s}$, so we conclude that $a^{(n-1)/2} \not\equiv -1 \pmod{n}$, and hence

$$\left(\frac{a}{n}\right) \not\equiv a^{(n-1)/2} \pmod{n}.$$

(d) If n is odd and composite, prove that $|G(n)| \leq (n-1)/2$.
Answer: This follows immediately from the results proven in parts (a), (b) and (c). If n is the product of distinct primes, then (c) shows that $G(n) \neq \mathbb{Z}_n{}^*$. Otherwise, (b) establishes the same result. Then the result shown in (a) can be applied.

(e) Summarize the above: prove that the error probability of the Solovay-Strassen primality test is at most $1/2$.
Answer: Suppose n is composite. If $\gcd(a,n) \neq 1$, then the Solovay-Strassen test returns the correct answer. If $\gcd(a,n) = 1$, then the Solovay-Strassen test returns the wrong answer if and only if $a \in G(n)$. We proved in part (d) that $|G(n)| \leq (n-1)/2$, so the probability of a wrong answer is at most

$$\frac{n-1-|\mathbb{Z}_n{}^*|}{n-1} \times 0 + \frac{|\mathbb{Z}_n{}^*|}{n-1} \times \frac{|G(n)|}{|\mathbb{Z}_n{}^*|} = \frac{|G(n)|}{n-1} \leq \frac{1}{2}.$$

5.23 Suppose we have a Las Vegas algorithm with failure probability ϵ.

(a) Prove that the probability of first achieving success on the nth trial is $p_n = \epsilon^{n-1}(1-\epsilon)$.
Answer: The probability of $n-1$ failures followed by a success is

$$\epsilon^{n-1}(1-\epsilon).$$

(b) The average (expected) number of trials to achieve success is

$$\sum_{n=1}^{\infty}(n \times p_n).$$

Show that this average is equal to $1/(1-\epsilon)$.

Answer:

$$\begin{aligned}\sum_{n=1}^{\infty}(n \times p_n) &= \sum_{n=1}^{\infty} n\epsilon^{n-1}(1-\epsilon)\\ &= (1-\epsilon)\sum_{n=1}^{\infty}\sum_{j=1}^{n}\epsilon^{n-1}\\ &= (1-\epsilon)\sum_{j=1}^{\infty}\sum_{n=j}^{\infty}\epsilon^{n-1}\\ &= (1-\epsilon)\sum_{j=1}^{\infty}\frac{\epsilon^{j-1}}{1-\epsilon}\\ &= \sum_{j=1}^{\infty}\epsilon^{j-1}\\ &= \frac{1}{1-\epsilon}.\end{aligned}$$

(c) Let δ be a positive real number less than 1. Show that the number of iterations required in order to reduce the probability of failure to at most δ is

$$\left\lfloor \frac{\log_2 \delta}{\log_2 \epsilon} \right\rfloor.$$

Answer: Note the number of iterations should be

$$\left\lceil \frac{\log_2 \delta}{\log_2 \epsilon} \right\rceil.$$

The probability of success after at most m trials is

$$\sum_{j=1}^{m} p_j = 1-\epsilon^m.$$

Therefore, the probability of failure after m trials is ϵ^m. We want to have $\epsilon^m \leq \delta$, which is equivalent to $m \log_2 \epsilon \leq \log_2 \delta$. Because $\log_2 \epsilon < 0$, this is the same as

$$m \geq \frac{\log_2 \delta}{\log_2 \epsilon}.$$

Since m is an integer, we require

$$m \geq \left\lceil \frac{\log_2 \delta}{\log_2 \epsilon} \right\rceil.$$

5.24 Suppose throughout this question that p is an odd prime and $\gcd(a,p)=1$.

(a) Suppose that $i \geq 2$ and $b^2 \equiv a \pmod{p^{i-1}}$. Prove that there is a unique $x \in \mathbb{Z}_{p^i}$ such that $x^2 \equiv a \pmod{p^i}$ and $x \equiv b \pmod{p^{i-1}}$. Describe how this x can be computed efficiently.

Answer: Since $b^2 \equiv a \pmod{p^{i-1}}$, we have that $b^2 - a = Kp^{i-1}$ for some integer K. Since $x \equiv b \pmod{p^{i-1}}$, we can write $x = Lp^{i-1} + b$ for some integer L. Now we compute x^2:

$$\begin{aligned}x^2 &= (Lp^{i-1}+b)^2 = L^2p^{2i-2} + 2bLp^{i-1} + b^2\\ &= L^2p^{2i-2} + 2bLp^{i-1} + Kp^{i-1} + a.\end{aligned}$$

Therefore $x^2 \equiv p^{i-1}(2bL+K)+a \pmod{p^i}$, so $x^2 \equiv a \pmod{p^i}$ if and only if $2bL+K \equiv 0 \pmod{p}$. This is true if and only if $L \equiv -K(2b)^{-1} \pmod{p}$.

(b) Illustrate your method in the following situation: starting with the congruence $6^2 \equiv 17 \pmod{19}$, find square roots of 17 modulo 19^2 and modulo 19^3.

Answer: $6^2 - 17 = 1 \times 19$, so $K = 1$. Then

$$L = -1 \times 12^{-1} \bmod 19 = -1 \times 8 \bmod 19 = 11,$$

so

$$x \equiv 11 \times 19 + 6 \equiv 215 \pmod{361}.$$

Next, $215^2 - 17 = 128 \times 361$, so $K = 128$. Then

$$L = -128 \times 8 \bmod 19 = 2$$

(there is no need to recalculate $(2b)^{-1} \bmod p$), so

$$x \equiv 2 \times 361 + 215 \pmod{19^3} \equiv 937 \pmod{19^3}.$$

(c) For all $i \geq 1$, prove that the number of solutions to the congruence $x^2 \equiv a \pmod{p^i}$ is either 0 or 2.

Answer: The proof is by induction on i. For $i = 1$, the congruence $x^2 \equiv a \pmod{p}$ has no solutions or two solutions in $\mathbb{Z}_p$, depending on the value of the Legendre symbol $\left(\frac{a}{p}\right)$. The result proved in part (a) establishes that the number of solutions modulo p^i is the same as the number of solutions modulo p^{i-1}, for all $i \geq 2$, so the result follows by induction.

5.25 Using various choices for the bound, B, attempt to factor 262063 and 9420457 using the $p-1$ method. How big does B have to be in each case to be successful?

Answer: When $n = 262063$, the factor 521 is computed when $B = 13$, but not when $B = 12$. (Note that $262063 = 521 \times 503$ and $520 = 2^3 \times 5 \times 13$. This illustrates why $B = 13$ is sufficient to find the factor 521.)

When $n = 9420457$, the factor 2351 is computed when $B = 47$, but not when $B = 46$. (Note that $9420457 = 2351 \times 4007$ and $2350 = 2 \times 5^2 \times 47$. This illustrates why $B = 47$ is sufficient to find the factor 2351.)

5.26 Factor 262063, 9420457 and 181937053 using the POLLARD RHO ALGORITHM, if the function f is defined to be $f(x) = x^2 + 1$. How many iterations are needed to factor each of these three integers?

Answer: When $n = 262063$, we get

$$\begin{aligned} x_{35} &= 225384, \\ x_{70} &= 94604, \\ \gcd(225384 &- 94604, 262063) = 503, \text{ and} \\ n &= 503 \times 521. \end{aligned}$$

When $n = 9420457$, we get

$$\begin{aligned} x_{50} &= 4559325, \\ x_{100} &= 6376648, \\ \gcd(4559325 &- 6376648, 262063) = 2351, \text{ and} \\ n &= 2351 \times 4007. \end{aligned}$$

When $n = 181937053$, we get

$$\begin{aligned} &x_{165} = 2452153,\\ &x_{330} = 73737576,\\ &\gcd(2452153 - 73737576, 181937053) = 12391, \text{ and}\\ &n = 12391 \times 14683. \end{aligned}$$

5.27 Suppose we want to factor the integer $n = 256961$ using the RANDOM SQUARES ALGORITHM. Using the factor base

$$\{-1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31\},$$

test the integers $z^2 \bmod n$ for $z = 500, 501, \ldots,$ until a congruence of the form $x^2 \equiv y^2 \pmod{n}$ is obtained and the factorization of n is found.

Answer: The following factorizations over the factor base are obtained:

$$\begin{aligned} 503^2 \bmod n &= (-1) \times 2^4 \times 13 \times 19\\ 504^2 \bmod n &= (-1) \times 5 \times 19 \times 31\\ 505^2 \bmod n &= (-1) \times 2^4 \times 11^2\\ 507^2 \bmod n &= 2^3 \times 11\\ 511^2 \bmod n &= 2^6 \times 5 \times 13\\ 516^2 \bmod n &= 5 \times 11 \times 13^2\\ 519^2 \bmod n &= 2^4 \times 5^2 \times 31 \end{aligned}$$

The first dependence relation that is obtained is:

$$(503 \times 504 \times 511 \times 519)^2 \equiv (2^7 \times 5^2 \times 13 \times 19 \times 31)^2 \pmod{n}.$$

The expressions inside the parentheses simplify to give

$$75319^2 \equiv 91105^2 \pmod{n}.$$

then we compute $\gcd(75319 + 91105, n) = 293$ and $\gcd(75319 - 91105, n) = 877$, so $n = 293 \times 877$.

5.28 In the RANDOM SQUARES ALGORITHM, we need to test a positive integer $w \leq n - 1$ to see if it factors completely over the factor base $\mathcal{B} = \{p_1, \ldots, p_B\}$ consisting of the B smallest prime numbers. Recall that $p_B = m \approx 2^s$ and $n \approx 2^r$.

(a) Prove that this can be done using at most $B + r$ divisions of an integer having at most r bits by an integer having at most s bits.

Answer: Consider the following algorithm:

```
Algorithm: TRIALDIVIDE($w, p_1, \ldots, p_B$)

$w_0 \leftarrow w$
for $i \leftarrow 1$ to $B$
    do $e_i \leftarrow 0$
       while $p_i | w_0$
           do $w_0 \leftarrow w_0 / p_i$
              $e_i \leftarrow e_i + 1$
return $(e_1, \ldots, e_B, w_0)$
```

At the end of TRIALDIVIDE, we have that

$$w = {p_1}^{e_1} \times \cdots \times {p_B}^{e_B} \times w_0,$$

where w_0 is not divisible by any of $p_1, \ldots, p_B$. The number of divisions performed in the algorithm is

$$B + e_1 + \cdots + e_B.$$

We have that

$$n > w \geq {p_1}^{e_1} \times \cdots \times {p_B}^{e_B} > 2^{e_1 + \cdots + e_B},$$

so

$$e_1 + \cdots + e_B < \log_2 n \approx r.$$

Therefore the number of divisions is (approximately) at most $B + r$.

(b) Assuming that $r < m$, prove that the complexity of this test is $O(rsm)$.
Answer: Each division takes time $O(rs)$ (see page 191). Therefore the total time is $O(rs(B + r))$. However, $B < m$ and we are assuming that $r < m$. Therefore $B + r$ is $O(m)$ and the total time is $O(rsm)$.

5.29 In this exercise, we show that parameter generation for the *RSA Cryptosystem* should take care to ensure that $q - p$ is not too small, where $n = pq$ and $q > p$.

(a) Suppose that $q - p = 2d > 0$, and $n = pq$. Prove that $n + d^2$ is a perfect square.
Answer: $n + d^2 = pq + d^2 = p(p + 2d) + d^2 = (p + d)^2$.

(b) Given an integer n which is the product of two odd primes, and given a small positive integer d such that $n + d^2$ is a perfect square, show how this information can be used to factor n.
Answer: Suppose $n + d^2 = s^2$. Then $n = (s - d)(s + d)$.

(c) Use this technique to factor $n = 2189284635403183$.
Answer: $n + 9^2 = 46789792^2$, and so

$$n = (46789792 - 9)(46789792 + 9) = 46789783 \times 46789801.$$

5.30 Suppose Bob has carelessly revealed his decryption exponent to be $a = 14039$ in an *RSA Cryptosystem* with public key $n = 36581$ and $b = 4679$. Implement the randomized algorithm to factor n given this information. Test your algorithm with the "random" choices $w = 9983$ and $w = 13461$. Show all computations.
Answer: We have that $ab - 1 = 65688480 = 2^5 \times 2052765$, so $r = 2052765$.

When $w = 9983$, we get $v = 35039$ and $v^2 \equiv 1 \pmod{n}$, so the algorithm fails.

When $w = 13461$, we get $v = 11747$, $v^2 \equiv 8477 \pmod{n}$, $v^4 \equiv 14445 \pmod{n}$ and $v^8 \equiv 1 \pmod{n}$. Hence the algorithm succeeds:

$$\gcd(14445 + 1, n) = 233$$

is a factor of n.

5.31 If $q_1, \ldots, q_m$ is the sequence of quotients obtained in the applying the EUCLIDEAN ALGORITHM with input r_0, r_1, prove that the continued fraction $[q_1, \ldots, q_m] = r_0/r_1$.
Answer: From the Euclidean Algorithm, we have

$$\begin{aligned} r_0 &= q_1 r_1 + r_2 \\ r_1 &= q_2 r_2 + r_3 \\ &\vdots \\ r_{m-1} &= q_m r_m. \end{aligned}$$

We will prove, by reverse induction on j, that $[q_j, \ldots, q_m] = r_{j-1}/r_j$ for $1 \leq j \leq m$. The base case is $j = m$, where $[q_m] = q_m = r_{m-1}/r_m$. Assume the formula is true for $j = i+1$, and then prove it is true for $j = i$. From the Euclidean Algorithm, we have

$$r_{j-1} = q_j r_j + r_{j+1},$$

so

$$\begin{aligned} r_{j-1}/r_j &= q_j + r_{j+1}/r_j \\ &= q_j + 1/(r_j/r_{j+1}) \\ &= q_j + 1/[q_{j+1}, \ldots, q_m] \quad \text{(by induction)} \\ &= [q_j, q_{j+1}, \ldots, q_m]. \end{aligned}$$

By induction, the result is true for $1 \leq j \leq m$. Setting $j = 1$, we see that $r_0/r_1 = [q_1, ..., q_m]$, as desired.

5.32 Suppose that $n = 317940011$ and $b = 77537081$ in the *RSA Cryptosystem*. Using WIENER'S ALGORITHM, attempt to factor n.
Answer: The continued fraction expansion of b/n is

$$[0, 4, 9, 1, 19, 1, 1, 15, 3, 2, 3, 71, 3, 2].$$

The first few convergents are 0, $1/4$, $9/37$, $10/41$, etc. If we let $c = 10$ and $d = 41$, then we obtain the quadratic equation $x^2 - 37980x + 317940011 = 0$. This equation has roots 12457 and 25523, which are the factors of n.

5.33 Consider the modification of the *Rabin Cryptosystem* in which $e_K(x) = x(x + B) \bmod n$, where $B \in \mathbb{Z}_n$ is part of the public key. Supposing that $p = 199$, $q = 211$, $n = pq$ and $B = 1357$, perform the following computations.

(a) Compute the encryption $y = e_K(32767)$.
Answer: $32767(32767 + 1357) \bmod 41989 = 16027$.

(b) Determine the four possible decryptions of this given ciphertext y.
Answer: $B/2 \bmod n = 21673$ and $B^2/4 \bmod n = 29975$. The decryptions are $x = \sqrt{4013} - 21673 \bmod n$. To find one square root of 4013 modulo n, compute $4013^{(p+1)/4} \bmod p = 86$ and $4013^{(q+1)/4} \bmod q = 209$. Then use the Chinese remainder theorem to solve the system $c \equiv 86 \pmod p$, $c \equiv 209 \pmod q$, yielding $c \equiv 29538 \pmod n$. A second square root is obtained by using the Chinese remainder theorem to solve the system $c \equiv 86 \pmod p$, $c \equiv -209 \pmod q$, yielding $c \equiv 1479 \pmod n$. The other two square roots are the negatives (modulo n) of the first two square roots. Therefore we obtain four square roots, namely 29538, 12451, 1479 and 40510. The four decryptions of y are $x = 7865$, 32767, 21795 and 18837.

5.34 Prove Equations (5.3) and (5.4) relating the functions $half$ and $parity$.
Answer: Denote $y = e_K(x)$, where $0 \leq x < n$. First, suppose that $half(y) = 0$. Then $0 \leq x < n/2$, and hence $0 \leq 2x < n$. Then

$$e_K(2x) \equiv e_K(2)e_K(x) \equiv e_K(2)y \pmod n.$$

Therefore $parity(e_K(2)y \bmod n) = 0$, because $2x$ is even.

Conversely, suppose that $parity(e_K(2)y \bmod n) = 0$. This implies that $2x \bmod$

n is even, where $0 \leq x < n$. However,

$$2x \bmod n = \begin{cases} 2x & \text{if } 0 \leq x < n/2 \\ 2x - n & \text{if } n/2 \leq x < n. \end{cases}$$

Because n is odd, we see that $2x \bmod n$ is even if and only if $0 \leq x < n/2$. This implies that $half(y) = 0$.

Now we turn to the other identity. First, suppose that $parity(y) = 0$. Then x is even, and hence $0 \leq x/2 < n/2$, where $x/2$ is an integer. Then

$$e_K(x/2) \equiv e_K(2^{-1})e_K(x) \equiv e_K(2^{-1})y \pmod{n}.$$

Therefore $half(e_K(2^{-1})y \bmod n) = 0$.

Finally, suppose that $half(e_K(2^{-1})y \bmod n) = 0$. This implies that $0 \leq 2^{-1}x \bmod n < n/2$, where $0 \leq x < n$. However,

$$2^{-1}x \bmod n = \begin{cases} x/2 & \text{if } x \text{ is even} \\ (x+n)/2 & \text{if } x \text{ is odd.} \end{cases}$$

We see that $2^{-1}x \bmod n < n/2$ if and only if x is even. This implies that $parity(y) = 0$.

5.35 Prove that Cryptosystem 5.3 is not semantically secure against a chosen ciphertext attack. Given x_1, x_2, a ciphertext (y_1, y_2) that is an encryption of x_i ($i = 1$ or 2), and given a decryption oracle DECRYPT for Cryptosystem 5.3, describe an algorithm to determine whether $i = 1$ or $i = 2$. You are allowed to call the algorithm DECRYPT with any input except for the given ciphertext (y_1, y_2), and it will output the corresponding plaintext.

Answer: Choose a random value $z \neq 0$, define $y_3 = y_2 \oplus z$, and call DECRYPT(y_1, y_2). The oracle outputs a value x, where $y_3 = G(r) \oplus x$. But

$$y_3 = y_2 \oplus z = G(r) \oplus x_i \oplus z,$$

where $i = 1$ or 2. Therefore $x_i = x \oplus z$ where x and z are known, and, hence, it is easy to determine the correct value of i.

6

Public-key Cryptosystems Based on the Discrete Logarithm Problem

Exercises

6.1 Implement SHANKS' ALGORITHM for finding discrete logarithms in $\mathbb{Z}_p{}^*$, where p is prime and α is a primitive element modulo p. Use your program to find $\log_{106} 12375$ in $\mathbb{Z}_{24691}{}^*$ and $\log_6 248388$ in $\mathbb{Z}_{458009}{}^*$.

Answer: When $p = 12375$, we have $m = 158$. We find that $j = 141$, $i = 114$ and $\log_{106} 12375 = 22392$.

When $p = 458009$, we have $m = 677$. We find that $j = 625$, $i = 343$ and $\log_6 248388 = 232836$.

6.2 Describe how to modify SHANKS' ALGORITHM to compute the logarithm of β to the base α in a group G if it is specified ahead of time that this logarithm lies in the interval $[s, t]$, where s and t are integers such that $0 \leq s < t < n$, where n is the order of α. Prove that your algorithm is correct, and show that its complexity is $O(\sqrt{t-s})$.

Answer: Define $\gamma = \alpha^{-s}\beta$. Then $\log_\alpha \beta \in [s,t]$ if and only if $\log_\alpha \gamma \in [0, t-s]$. It suffices to compute $\log_\alpha \gamma$ using SHANKS' ALGORITHM with $m = \lceil \sqrt{t-s+1} \rceil$, and then calculate $\log_\alpha \beta = \log_\alpha \gamma + s$.

The proof of correctness is essentially the same as the proof of correctness of SHANKS' ALGORITHM given in Section 6.2.

The complexity of the algorithm is $O(\sqrt{t-s+1}) = O(\sqrt{t-s})$.

6.3 The integer $p = 458009$ is prime and $\alpha = 2$ has order 57251 in $\mathbb{Z}_p{}^*$. Use the POLLARD RHO ALGORITHM to compute the discrete logarithm in $\mathbb{Z}_p{}^*$of $\beta = 56851$ to the base α. Take the initial value $x_0 = 1$, and define the partition (S_1, S_2, S_3) as in Example 6.3. Find the smallest integer i such that $x_i = x_{2i}$, and then compute the desired discrete logarithm.

Answer: $x_{444} = 339768$, $a_{444} = 22811$, $b_{444} = 35067$, $x_{888} = 339768$, $a_{888} = 37251$ and $b_{888} = 5360$. Thereore, $\log_\alpha \beta = 40007$.

6.4 Suppose that p is an odd prime and k is a positive integer. The multiplicative group $\mathbb{Z}_{p^k}{}^*$ has order $p^{k-1}(p-1)$, and is known to be cyclic. A generator for this group is called a *primitive element modulo* p^k.

(a) Suppose that α is a primitive element modulo p. Prove that at least one of α

or $\alpha + p$ is a primitive element modulo p^2.

Answer: Suppose that α has order $p - 1$ in $\mathbb{Z}_p{}^*$. Let n denote the order of α in $Z_{p^2}{}^*$. $\alpha^n \equiv 1 \pmod{p^2}$ implies $\alpha^n \equiv 1 \pmod{p}$, so $n \equiv 0 \pmod{p-1}$. Also, n divides $|\mathbb{Z}_{p^2}{}^*| = p^2 - p$. Therefore $n = p - 1$ or $n = p^2 - p$. If $n = p^2 - p$, then we're done, so assume $n = p - 1$. Now consider $\alpha + p$. By the same argument, α has order $p - 1$ or $p^2 - p$ in $\mathbb{Z}_{p^2}{}^*$. We show that $\alpha + p$ cannot have order $p - 1$, which finishes the proof.

We expand $(\alpha + p)^{p-1}$ using the binomial theorem:

$$(\alpha + p)^{p-1} = \alpha^{p-1} + (p-1)\alpha^{p-2}p + \text{ terms divisible by } p^2.$$

Reducing modulo p^2, we see that

$$\begin{aligned}(\alpha + p)^{p-1} &\equiv \alpha^{p-1} - p\alpha^{p-2} \pmod{p^2}\\ &\equiv 1 - p\alpha^{p-2} \pmod{p^2}.\end{aligned}$$

Therefore $(\alpha + p)^{p-1} \equiv 1 \pmod{p^2}$ if and only if $\alpha^{p-2} \equiv 0 \pmod{p}$. However, $\gcd(\alpha, p) = 1$. Therefore, $\alpha + p$ does not have order $p - 1$, and we're done.

(b) Describe how to efficiently verify that 3 is a primitive root modulo 29 and modulo 29^2. Note: It can be shown that if α is a primitive root modulo p and modulo p^2, then it is a primitive root modulo p^k for all positive integers k (you do not have to prove this fact). Therefore, it follows that 3 is a primitive root modulo 29^k for all positive integers k.

Answer: $28 = 2^2 7$. To show that 3 is primitive modulo 29, it suffices to show that $3^{28/2}$ and $3^{28/7}$ are not congruent to 1 modulo 29. Since $3^1 4 \equiv 28 \pmod{29}$ and $3^4 \equiv 23 \pmod{29}$, we conclude that 3 is primitive modulo 29.

As shown in (a), the order of 3 in $\mathbb{Z}_{29^2}{}^*$ is either 28 or 28×29. To show that 3 is a primitive element, it suffices to show that 3^{28} is not congruent to 1 modulo 29^2. Since $3^{28} \bmod (29^2) = 436$, we're done.

(c) Find an integer α that is a primitive root modulo 29 but not a primitive root modulo 29^2.

Answer: It suffices to find a value α such that $\alpha^{28/2}$ and $\alpha^{28/7}$ are not congruent to 1 modulo 29; but $\alpha^{28} \equiv 1 \pmod{29^2}$. We have $14^{14} \equiv 28 \pmod{19}$, $14^4 \equiv 20 \pmod{29}$ and $14^{28} \equiv 1 \pmod{29^2}$, so $\alpha = 14$ is such an integer.

(d) Use the POHLIG-HELLMAN ALGORITHM to compute the discrete logarithm of 3344 to the base 3 in the multiplicative group $\mathbb{Z}_{24389}{}^*$.

Answer: $|\mathbb{Z}_{24389}{}^*| = 29^2 28 = 29^2 2^2 7^1$. Let a denote the desired discrete logarithm. We need to compute $a \bmod 29^2$, $a \bmod 2^2$ and $a \bmod 7$. We obtain:

$$\begin{aligned}a &\equiv 260 \pmod{29^2}\\ a &\equiv 2 \pmod{2^2}\\ a &\equiv 2 \pmod{7}.\end{aligned}$$

Applying the Chinese remainder theorem, $a = 18762$.

6.5 Implement the POHLIG-HELLMAN ALGORITHM for finding discrete logarithms in $\mathbb{Z}_p$, where p is prime and α is a primitive element. Use your program to find

$\log_5 8563$ in $\mathbb{Z}_{28703}$ and $\log_{10} 12611$ in $\mathbb{Z}_{31153}$.

Answer: $28702 = 2^1 113^1 127^1$. We find that

$$\log_5 8563 \equiv 1 \pmod{2},$$

$$\log_5 8563 \equiv 67 \pmod{113}, \quad \text{and}$$

$$\log_5 8563 \equiv 99 \pmod{127}.$$

Using the Chinese remainder theorem, $\log_5 8563 = 3909$.

$31152 = 2^4 3^1 11^1 59^1$. We find that

$$\log_{10} 12611 \equiv 14 \pmod{16},$$

$$\log_{10} 12611 \equiv 2 \pmod{3},$$

$$\log_{10} 12611 \equiv 8 \pmod{11}, \quad \text{and}$$

$$\log_{10} 12611 \equiv 51 \pmod{59}.$$

Using the Chinese remainder theorem, $\log_{10} 12611 = 17102$.

6.6 Let $p = 227$. The element $\alpha = 2$ is primitive in $\mathbb{Z}_p{}^*$.

(a) Compute α^{32}, α^{40}, α^{59} and α^{156} modulo p, and factor them over the factor base $\{2, 3, 5, 7, 11\}$.

Answer: $2^{32} \equiv 176 = 2^4 \times 11$, $2^{40} \equiv 110 = 2 \times 5 \times 11$, $2^{59} \equiv 60 = 2^2 \times 3 \times 5$ and $2^{156} \equiv 28 = 2^2 \times 7$

(b) Using the fact that $\log 2 = 1$, compute $\log 3$, $\log 5$, $\log 7$ and $\log 11$ from the factorizations obtained above (all logarithms are discrete logarithms in $\mathbb{Z}_p{}^*$ to the base α).

Answer: $\log 2 = 1$, $\log 3 = 46$, $\log 5 = 11$, $\log 7 = 154$ and $\log 11 = 28$.

(c) Now suppose we wish to compute $\log 173$. Multiply 173 by the "random" value $2^{177} \bmod p$. Factor the result over the factor base, and proceed to compute $\log 173$ using the previously computed logarithms of the numbers in the factor base.

Answer: $173 \times 2^{177} \equiv 168 = 2^3 3^1 7^1$. Therefore, $\log 173 = 2 \log 2 + \log 3 + \log 7 - 177 \bmod 226 = 26$.

6.7 Suppose that $n = pq$ is an RSA modulus (i.e., p and q are distinct odd primes), and let $\alpha \in \mathbb{Z}_n{}^*$. For a positive integer m and for any $\alpha \in \mathbb{Z}_m{}^*$, define $\mathsf{ord}_m(\alpha)$ to be the order of α in the group $\mathbb{Z}_m{}^*$.

(a) Prove that

$$\mathsf{ord}_n(\alpha) = \mathrm{lcm}(\mathsf{ord}_p(\alpha), \mathsf{ord}_q(\alpha)).$$

Answer: This follows because $\alpha^j \equiv 1 \pmod{n}$ if and only if $\alpha^j \equiv 1 \pmod{p}$ and $\alpha^j \equiv 1 \pmod{q}$.

(b) Suppose that $\gcd(p-1, q-1) = d$. Show that there exists an element $\alpha \in \mathbb{Z}_n{}^*$ such that

$$\mathsf{ord}_n(\alpha) = \frac{\phi(n)}{d}.$$

Answer: Let α_p be a primitive element modulo p and let α_q be a primitive element modulo q. Using the Chinese remainder theorem, there exists $\alpha \in \mathbb{Z}_n{}^*$ such that $\alpha \equiv \alpha_p \pmod{p}$ and $\alpha \equiv \alpha_q \pmod{q}$. Then $\mathsf{ord}_p(\alpha) = p - 1$ and $\mathsf{ord}_q(\alpha) = q - 1$. Applying the result proven in part (a), we have that

$$\mathsf{ord}_n(\alpha) = \mathrm{lcm}(p-1, q-1) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)} = \frac{\phi(n)}{d}.$$

(c) Suppose that $\gcd(p-1, q-1) = 2$, and we have an oracle that solves the Discrete Logarithm problem in the subgroup $\langle\alpha\rangle$, where $\alpha \in {\mathbb{Z}_n}^*$ has order $\phi(n)/2$. That is, given any $\beta \in \langle\alpha\rangle$, the oracle will find the discrete logarithm $a = \log_\alpha \beta$, where $0 \leq a \leq \phi(n)/2 - 1$. (The value $\phi(n)/2$ is secret however.) Suppose we compute the value $\beta = \alpha^n \bmod n$ and then we use the oracle to find $a = \log_\alpha \beta$. Assuming that $p > 3$ and $q > 3$, prove that $n - a = \phi(n)$.

Answer: Because $\alpha^n \equiv \alpha^a \pmod{n}$ and α has order $\phi(n)/2$, we have that $a = n - k\phi(n)/2$ for some integer k. Also, $0 \leq a < \phi(n)/2$, so there is a unique integer k such that $0 \leq n - k\phi(n)/2 < \phi(n)/2$. We will show that $k = 2$ causes this inequality to be satisfied, which will complete the proof.

When $k = 2$, the inequality is equivalent to the following:

$$\phi(n) \leq n < \frac{3\phi(n)}{2}.$$

Clearly $\phi(n) \leq n$, so it suffices to show that $2n < 3\phi(n)$. Assuming WLOG that $p > q$, and using the fact that $q > 3$, this is equivalent to the following:

$$\begin{aligned} 2pq &< 3(p-1)(q-1), \\ pq &> 3(p+q-1), \\ p &> 3 + \frac{6}{q-3}. \end{aligned}$$

Because $q \geq 5$, we have that $3 + \frac{6}{q-3} \leq 6$. However, $p > q \geq 5$ is prime, so $p \geq 7$, and therefore the inequality is satisfied.

(d) Describe how n can easily be factored, given the discrete logarithm $a = \log_\alpha \beta$ from (c).

Answer: Given a, it is simple to compute $\phi(n) = n - a$. Then, given n and $\phi(n)$, it is straightforward to factor n by solving a quadratic equation, as described in Section 5.7.1.

6.8 In this question, we consider a generic algorithm for the Discrete Logarithm problem in $(\mathbb{Z}_{19}, +)$.

(a) Suppose that the set C is defined as follows:

$$C = \{(1 - i^2 \bmod 19, i \bmod 19) : i = 0, 1, 2, 4, 7, 12\}.$$

Compute $\mathsf{Good}(C)$.

Answer: observe that

$$\frac{1 - i^2 - (1 - j^2)}{i - j} = -(i + j),$$

for any $i \neq j$. From this it follows that

$$\mathsf{Good}(C) = \{i + j \bmod 19 : i, j \in \{0, 1, 2, 4, 7, 12\}, i \neq j\}.$$

An easy computation then shows that

$$\mathsf{Good}(C) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 16\}.$$

(b) Suppose that the output of the group oracle, given the ordered pairs in C, is as follows:

$$\begin{aligned}(0,1) &\mapsto 10111\\ (1,0) &\mapsto 01100\\ (16,2) &\mapsto 00110\\ (4,4) &\mapsto 01010\\ (9,7) &\mapsto 00100\\ (9,12) &\mapsto 11001,\end{aligned}$$

where group elements are encoded as (random) binary 5-tuples. What can you say about the value of "a"?

Answer: Because the encodings are all different, it must be the case that $a \notin \mathsf{Good}(C)$. Therefore $a = 10, 15, 17$ or 18.

6.9 Decrypt the ElGamal ciphertext presented in Table 6.3. 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 5.12.

The plaintext was taken from "The English Patient," by Michael Ondaatje, Alfred A. Knopf, Inc., New York, 1992.

Answer: The first ciphertext element, $(3781, 14409)$, is decrypted to the plaintext element

$$x = 14409((3781)^{7899})^{-1} \bmod 31847 = 12354.$$

$x = 12354$ encodes the triple $18, 7, 4$, which corresponds to the three letters s, h, e.

The complete plaintext is as follows:

She stands up in the garden where she has been working and looks into the distance. She has sensed a change in the weather. There is another gust of wind, a buckle of noise in the air, and the tall cypresses sway. She turns and moves uphill towards the house. Climbing over a low wall, feeling the first drops of rain on her bare arms, she crosses the loggia and quickly enters the house.

6.10 Determine which of the following polynomials are irreducible over $\mathbb{Z}_2[x]$: $x^5 + x^4 + 1$, $x^5 + x^3 + 1$, $x^5 + x^4 + x^2 + 1$.

Answer: $x^5 + x^3 + 1$ is irreducible, $x^5 + x^4 + 1 = (x^3 + x + 1)(x^2 + x + 1)$ and $x^5 + x^4 + x^2 + 1 = (x + 1)(x^4 + x + 1)$.

6.11 The field $\mathbb{F}_{2^5}$ can be constructed as $\mathbb{Z}_2[x]/(x^5 + x^2 + 1)$. Perform the following computations in this field.

(a) Compute $(x^4 + x^2) \times (x^3 + x + 1)$.

Answer: In the ring $\mathbb{Z}_2[x]$, we have that

$$(x^4 + x^2) \times (x^3 + x + 1) = x^2(x^5 + x^2 + 1) + x^3,$$

so $(x^4 + x^2) \times (x^3 + x + 1) = x^3$ in the field $\mathbb{Z}_2[x]/(x^5 + x^2 + 1)$.

(b) Using the extended Euclidean algorithm, compute $(x^3 + x^2)^{-1}$.

Answer: $(x^3 + x^2)^{-1} = x^2 + x + 1$ in the field $\mathbb{Z}_2[x]/(x^5 + x^2 + 1)$.

(c) Using the square-and-multiply algorithm, compute x^{25}.

Answer: $x^{25} = x^4 + x^3 + 1$ in the field $\mathbb{Z}_2[x]/(x^5 + x^2 + 1)$.

6.12 We give an example of the *ElGamal Cryptosystem* implemented in $\mathbb{F}_{3^3}$. The polynomial $x^3 + 2x^2 + 1$ is irreducible over $\mathbb{Z}_3[x]$ and hence $\mathbb{Z}_3[x]/(x^3 + 2x^2 + 1)$

TABLE 6.3
ElGamal Ciphertext

$(3781, 14409)$	$(31552, 3930)$	$(27214, 15442)$	$(5809, 30274)$
$(5400, 31486)$	$(19936, 721)$	$(27765, 29284)$	$(29820, 7710)$
$(31590, 26470)$	$(3781, 14409)$	$(15898, 30844)$	$(19048, 12914)$
$(16160, 3129)$	$(301, 17252)$	$(24689, 7776)$	$(28856, 15720)$
$(30555, 24611)$	$(20501, 2922)$	$(13659, 5015)$	$(5740, 31233)$
$(1616, 14170)$	$(4294, 2307)$	$(2320, 29174)$	$(3036, 20132)$
$(14130, 22010)$	$(25910, 19663)$	$(19557, 10145)$	$(18899, 27609)$
$(26004, 25056)$	$(5400, 31486)$	$(9526, 3019)$	$(12962, 15189)$
$(29538, 5408)$	$(3149, 7400)$	$(9396, 3058)$	$(27149, 20535)$
$(1777, 8737)$	$(26117, 14251)$	$(7129, 18195)$	$(25302, 10248)$
$(23258, 3468)$	$(26052, 20545)$	$(21958, 5713)$	$(346, 31194)$
$(8836, 25898)$	$(8794, 17358)$	$(1777, 8737)$	$(25038, 12483)$
$(10422, 5552)$	$(1777, 8737)$	$(3780, 16360)$	$(11685, 133)$
$(25115, 10840)$	$(14130, 22010)$	$(16081, 16414)$	$(28580, 20845)$
$(23418, 22058)$	$(24139, 9580)$	$(173, 17075)$	$(2016, 18131)$
$(19886, 22344)$	$(21600, 25505)$	$(27119, 19921)$	$(23312, 16906)$
$(21563, 7891)$	$(28250, 21321)$	$(28327, 19237)$	$(15313, 28649)$
$(24271, 8480)$	$(26592, 25457)$	$(9660, 7939)$	$(10267, 20623)$
$(30499, 14423)$	$(5839, 24179)$	$(12846, 6598)$	$(9284, 27858)$
$(24875, 17641)$	$(1777, 8737)$	$(18825, 19671)$	$(31306, 11929)$
$(3576, 4630)$	$(26664, 27572)$	$(27011, 29164)$	$(22763, 8992)$
$(3149, 7400)$	$(8951, 29435)$	$(2059, 3977)$	$(16258, 30341)$
$(21541, 19004)$	$(5865, 29526)$	$(10536, 6941)$	$(1777, 8737)$
$(17561, 11884)$	$(2209, 6107)$	$(10422, 5552)$	$(19371, 21005)$
$(26521, 5803)$	$(14884, 14280)$	$(4328, 8635)$	$(28250, 21321)$
$(28327, 19237)$	$(15313, 28649)$		

is the field $\mathbb{F}_{3^3}$. We can associate the 26 letters of the alphabet with the 26 nonzero field elements, and thus encrypt ordinary text in a convenient way. We will use a lexicographic ordering of the (nonzero) polynomials to set up the correspondence. This correspondence is as follows:

A	$\leftrightarrow$	1	B	$\leftrightarrow$	2	C	$\leftrightarrow$	x
D	$\leftrightarrow$	$x+1$	E	$\leftrightarrow$	$x+2$	F	$\leftrightarrow$	$2x$
G	$\leftrightarrow$	$2x+1$	H	$\leftrightarrow$	$2x+2$	I	$\leftrightarrow$	x^2
J	$\leftrightarrow$	x^2+1	K	$\leftrightarrow$	x^2+2	L	$\leftrightarrow$	x^2+x
M	$\leftrightarrow$	x^2+x+1	N	$\leftrightarrow$	x^2+x+2	O	$\leftrightarrow$	x^2+2x
P	$\leftrightarrow$	x^2+2x+1	Q	$\leftrightarrow$	x^2+2x+2	R	$\leftrightarrow$	$2x^2$
S	$\leftrightarrow$	$2x^2+1$	T	$\leftrightarrow$	$2x^2+2$	U	$\leftrightarrow$	$2x^2+x$
V	$\leftrightarrow$	$2x^2+x+1$	W	$\leftrightarrow$	$2x^2+x+2$	X	$\leftrightarrow$	$2x^2+2x$
Y	$\leftrightarrow$	$2x^x+2x+1$	Z	$\leftrightarrow$	$2x^x+2x+2$			

Suppose Bob uses $\alpha = x$ and $a = 11$ in an *ElGamal Cryptosystem*; then $\beta = x+2$. Show how Bob will decrypt the following string of ciphertext:

```
(K,H)(P,X)(N,K)(H,R)(T,F)(V,Y)(E,H)(F,A)(T,W)(J,D)(U,J)
```

Answer: The plaintext is $Galois\ field$.

6.13 Let E be the elliptic curve $y^2 = x^3 + x + 28$ defined over $\mathbb{Z}_{71}$.

(a) Determine the number of points on E.

Answer: $\#E = 72$.

(b) Show that E is not a cyclic group.

Answer: This follows from part (c). If E were cyclic, there would be points having order 72, but there are no such points.

Alternatively, the result proven in Exercise 6.14 can be applied, because the congruence $x^3 + x + 28 \equiv 0 \pmod{71}$ has three solutions (namely, $x = 27, 53$ and 62).

(c) What is the maximum order of an element in E? Find an element having this order.

Answer: The maximum order of a point is 36; $(4, 5)$ is one point having order 36. (E is isomorphic to $\mathbb{Z}_{36} \times \mathbb{Z}_2$.)

6.14 Suppose that $p > 3$ is an odd prime, and $a, b \in \mathbb{Z}_p$. Further, suppose that the equation $x^3 + ax + b \equiv 0 \pmod{p}$ has three distinct roots in $\mathbb{Z}_p$. Prove that the corresponding elliptic curve group $(E, +)$ is not cyclic.

HINT Show that the points of order two generate a subgroup of $(E, +)$ that is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Answer: Let a_1, a_2 and a_3 be the three roots, which must be distinct. It is easy to show that $x_1 = (a_1, 0)$, $x_2 = (a_2, 0)$ and $x_3 = (a_3, 0)$ are three distinct points on E having order 2.

Using the fact that $a_1 + a_2 + a_3 = 0$ (which follows because the coefficient of x^2 in the cubic equation $x^3 + ax + b = 0$ is 0), it is straightforward to show that $x_1 + x_2 = x_3$, $x_1 + x_3 = x_2$ and $x_2 + x_3 = x_1$. Hence $\{x_1, x_2, x_3, \mathcal{O}\}$ is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$. Since G contains a subgroup that is not cyclic, G is not cyclic.

6.15 Consider an elliptic curve E described by the formula $y^2 \equiv x^3 + ax + b \pmod{p}$, where $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ and $p > 3$ is prime.

(a) It is clear that a point $P = (x_1, y_1) \in E$ has order 3 if and only if $2P = -P$. Use this fact to prove that, if $P = (x_1, y_1) \in E$ has order 3, then

$$3{x_1}^4 + 6a{x_1}^2 + 12x_1 b - a^2 \equiv 0 \pmod{p}. \tag{6.7}$$

Answer: The x-coordinate of $-P$ is x_1. The x-coordinate of $2P$ is

$$\frac{(3{x_1}^2 + a)^2}{(2y_1)^2} - 2x_1.$$

These two x-coordinates must be equal if $P = -2P$. Hence,

$$(3{x_1}^2 + a)^2 = (2y_1)^2(3x_1) = 12x_1{y_1}^2.$$

However,

$${y_1}^2 = {x_1}^3 + ax_1 + b,$$

so

$$(3{x_1}^2 + a)^2 = 12x_1({x_1}^3 + ax_1 + b).$$

This simplifies to give the equation (6.7), which is a necessary condition for P to have order 3.

(b) Conclude from equation (6.7) that there are at most 8 points of order 3 on the elliptic curve E.

Answer: (6.7) is a fourth degree equation, which has at most four roots over the field $\mathbb{Z}_p$. For each root x_1 of (6.7), there are at most two values of y_1 such that (x_1, y_1) is a point on E. The total number of points on E having order 3 is therefore at most $2 \times 4 = 8$.

(c) Using equation (6.7), determine all points of order 3 on the elliptic curve $y^2 \equiv x^3 + 34x \pmod{73}$.

Answer: The equation (6.7) becomes

$$
\begin{aligned}
3{x_1}^4 + 6 \times 34{x_1}^2 - 34^2 &\equiv 0 \pmod{73},\\
{x_1}^4 + 68{x_1}^2 - 69 &\equiv 0 \pmod{73},\\
{x_1}^4 - 5{x_1}^2 + 4 &\equiv 0 \pmod{73}.
\end{aligned}
$$

This equation factors:

$$({x_1}^2 - 1)({x_1}^2 - 4) \equiv 0 \pmod{73},$$

or

$$x_1 \equiv 1, -1, 2, -2 \pmod{73}.$$

For each of these values of x_1, we need to find the corresponding values of y_1 (if possible).

i. If $x_1 = 1$, then ${y_1}^2 = 35$, and $y_1 = 20$ or 53.

ii. If $x_1 = -1$, then ${y_1}^2 = 38$, and $y_1 = 29$ or 35.

iii. If $x_1 = 2$, then ${y_1}^2 = 2$, and $y_1 = 21$ or 52.

iv. If $x_1 = -2$, then ${y_1}^2 = 71$, and $y_1 = 17$ or 56.

There are eight possible points of order 3, namely $(1, 20)$, $(1, 53)$, $(72, 29)$, $(72, 44)$, $(2, 21)$, $(2, 52)$, $(71, 17)$ and $(71, 56)$. (It can be verified that all eight of these points do in fact have order 3.)

6.16 Suppose that E is an elliptic curve defined over $\mathbb{Z}_p$, where $p > 3$ is prime. Suppose that $\#E$ is prime, $P \in E$, and $P \neq \mathcal{O}$.

(a) Prove that the discrete logarithm $\log_P(-P) = \#E - 1$.

Answer: Denote $\#E = q$. The order of P divides q, $P \neq \mathcal{O}$ and q is prime, so the order of P must be equal to q. Now we have that $\mathcal{O} = qP = P + (q-1)P$. But we also have $\mathcal{O} = P + (-P)$, so $-P = (q-1)P$ and $\log_P(-P) = q - 1$.

(b) Describe how to compute $\#E$ in time $O(p^{1/4})$ by using Hasse's bound on $\#E$, together with a modification of SHANKS' ALGORITHM. Give a pseudocode description of the algorithm.

Answer: Let $P \in E$, $P \neq \mathcal{O}$. Define $s = \lfloor p - \sqrt{p} \rfloor$ and $t = \lfloor p + \sqrt{p} \rfloor - 1$, and use the modification of SHANKS' ALGORITHM described in Exercise 6.2 to find $a = \log_P(-P)$. (We have that $\log_P(-P) = q - 1$, where $q \in [s+1, t+1]$, so $\log_P(-P) \in [s, t]$.)

Note that the interval $[s, t]$ contains $2\lfloor \sqrt{p} \rfloor + 1$ possible values. It will be the case that $q = a + 1$ provided that $q \geq 2\lfloor \sqrt{p} \rfloor + 1$ (this ensures that there is a unique element of the interval $[s, t]$ that is congruent to a modulo q). We have that $q \geq p - \lfloor \sqrt{p} \rfloor$, so everything is all right, provided that $p \geq 3\lfloor \sqrt{p} \rfloor + 1$. This last inequality is true for all primes $p \geq 11$.

For the primes $p = 5$ and 7, it is probably simpler to directly compute the value of q. This does not affect the asymptotic complexity of the algorithm, which is $O(\sqrt{t-s}) = O(p^{1/4})$ by Exercise 6.2.

6.17 Let E be the elliptic curve $y^2 = x^3 + 2x + 7$ defined over $\mathbb{Z}_{31}$. It can be shown that $\#E = 39$ and $P = (2, 9)$ is an element of order 39 in E. The *Simplified ECIES* defined on E has $\mathbb{Z}_{31}{}^*$ as its plaintext space. Suppose the private key is $m = 8$.

(a) Compute $Q = mP$.
Answer: $8P = (8, 15)$.

(b) Decrypt the following string of ciphertext:

$$((18, 1), 21), ((3, 1), 18), ((17, 0), 19), ((28, 0), 8).$$

Answer: The plaintext is $20\ 9\ 12\ 5$.

(c) Assuming that each plaintext represents one alphabetic character, convert the plaintext into an English word. (Here we will use the correspondence $A \leftrightarrow 1$, ..., $Z \leftrightarrow 26$, because 0 is not allowed in a (plaintext) ordered pair.)
Answer: $tile$

6.18 (a) Determine the NAF representation of the integer 87.
Answer: The NAF representation of 87 is $(1, 0, -1, 0, -1, 0, 0, -1)$.

(b) Using the NAF representation of 87, use Algorithm 6.5 to compute $87P$, where $P = (2, 6)$ is a point on the elliptic curve $y^2 = x^3 + x + 26$ defined over $\mathbb{Z}_{127}$. Show the partial results during each iteration of the algorithm.
Answer: The algorithm proceeds as follows:

i	c_i	Q
7	1	$[2, 6]$
6	0	$[118, 80]$
5	-1	$[68, 57]$
4	0	$[85, 119]$
3	-1	$[87, 116]$
2	0	$[91, 18]$
1	0	$[102, 39]$
0	-1	$[102, 88]$

Therefore $87P = [102, 88]$.

6.19 Let $\mathcal{L}_i$ denote the set of positive integers that have exactly i coefficients in their NAF representation, such that the leading coefficient is 1. Denote $k_i = |\mathcal{L}_i|$.

(a) By means of a suitable decomposition of $\mathcal{L}_i$, prove that the k_i's satisfy the following recurrence relation:

$$k_1 = 1$$

$$k_2 = 1$$

$$k_{i+1} = 2(k_1 + k_2 + ... + k_{i-1}) + 1 \quad \text{(for } i \geq 2\text{)}.$$

Answer: It is clear that $k_1 = k_2 = 1$.

For any $x \in \mathcal{L}_{i+1}$, let $a(x)$ denote the number of consecutive zeroes that follow the initial '1'. If $a(x) = i$, then the NAF representation of x is $(1, 0, \ldots, 0)$. If $a(x) \leq i - 1$, then let y denote the entry that follows the $a(x)$ consecutive zeroes in the NAF representation of x. Clearly $y = 1$ or $y = -1$. If $y = 1$, then the last $i - a(x)$ entries in the NAF representation of x form the NAF representation of an integer in $\mathcal{L}_{i-a(x)}$. Suppose that $y = -1$. If we change this '-1' to a '1', then the last $i - a(x)$ entries again form the NAF representation of an integer in $\mathcal{L}_{i-a(x)}$.

(b) Derive a second degree recurrence relation for the k_i's, and obtain an explicit solution of the recurrence relation.
Answer: We have that

$$k_{i+1} = 2(k_1 + k_2 + \dots + k_{i-1}) + 1$$

and

$$k_i = 2(k_1 + k_2 + \dots + k_{i-2}) + 1,$$

for $i \geq 3$. Subtracting, we see that

$$k_{i+1} - k_i = 2k_{i-1},$$

$i \geq 3$. Also, $k_1 = k_2 = 1$ and $k_3 = 3$.

This recurrence can be solved by standard techniques; the solution is

$$k_i = \frac{2^{i+1}}{3} + \frac{(-1)^i}{3},$$

$i \geq 1$ (this can be proven by induction).

6.20 Find $\log_5 896$ in $\mathbb{Z}_{1103}$ using Algorithm 6.6, given that $L_2(\beta) = 1$ for $\beta = 25, 219$ and 841, and $L_2(\beta) = 0$ for $\beta = 163, 532, 625$ and 656.
Answer: We obtain the following:

$$\begin{array}{ll} \beta = 896 & x_0 = 1 \\ \beta = 841 & x_1 = 1 \\ \beta = 656 & x_2 = 0 \\ \beta = 532 & x_3 = 0 \\ \beta = 219 & x_4 = 1 \\ \beta = 163 & x_5 = 0 \\ \beta = 625 & x_6 = 0 \\ \beta = 25 & x_7 = 1 \\ \beta = 1 & \end{array}$$

Therefore $\log_5 896 = 10010011_2 = 147$.

6.21 Throughout this question, suppose that $p \equiv 5 \pmod 8$ is prime and suppose that a is a quadratic residue modulo p.

(a) Prove that $a^{(p-1)/4} \equiv \pm 1 \pmod p$.
Answer: a is a quadratic residue, so $a^{(p-1)/2} \equiv 1 \pmod p$ by Euler's criterion. Now $(a^{(p-1)/4})^2 = a^{(p-1)/2}$, so $a^{(p-1)/4} \equiv \pm 1 \pmod p$.

(b) If $a^{(p-1)/4} \equiv 1 \pmod p$, prove that $a^{(p+3)/8} \bmod p$ is a square root of a modulo p.
Answer:

$$(a^{(p+3)/8})^2 = a^{(p+3)/4} = a^{(p-1)/4}a \equiv a^{(p-1)/4} \pmod p.$$

(c) If $a^{(p-1)/4} \equiv -1 \pmod p$, prove that $2^{-1}(4a)^{(p+3)/8} \bmod p$ is a square root of a modulo p.

HINT Use the fact that $\left(\frac{2}{p}\right) = -1$ when $p \equiv 5 \pmod 8$ is prime.

Answer:

$$\begin{aligned}(2^{-1}(4a)^{(p+3)/8})^2 &\equiv 4^{-1}(4a)^{(p+3)/4} \pmod p\\ &\equiv 4^{(p-1)/4}a^{(p+3)/4} \pmod p\\ &\equiv 2^{(p-1)/2}a^{(p-1)/4}a \pmod p\\ &\equiv \left(\frac{2}{p}\right)a^{(p-1)/4}a \pmod p\\ &\equiv (-1)(-1)a \pmod p\\ &\equiv a \pmod p.\end{aligned}$$

(d) Given a primitive element $\alpha \in \mathbb{Z}_p{}^*$, and given any $\beta \in \mathbb{Z}_p{}^*$, show that $L_2(\beta)$ can be computed efficiently.

HINT Use the fact that it is possible to compute square roots modulo p, as well as the fact that $L_1(\beta) = L_1(p - \beta)$ for all $\beta \in \mathbb{Z}_p{}^*$, when $p \equiv 5 \pmod 8$ is prime.

Answer: Let $a = \log_\alpha \beta$. Then $a = 4a^* + 2a_1 + a_0$, where $a_0 = L_1(\beta)$ and $a_1 = L_2(\beta)$. It is possible to compute a_0 efficiently (see page 262). Let $\beta_0 = \beta/\alpha^{a_0}$; then $\log_\alpha \beta_0 = 4a^* + 2a_1$. Next, compute a square root β_1 of β_0 using the technique described in part (b) or (c). The two square roots of β_0 are $\pm\alpha^{2a^*+a_1}$, or $\alpha^{2a^*+a_1}$ and $\alpha^{(p-1)/2-2a^*-a_1}$. We have that $(p-1)/2$ is even, so $L_1(\alpha^{2a^*+a_1}) = L_1(\alpha^{(p-1)/2-2a^*-a_1})$. Therefore,

$$L_1(\beta_1) = L_1(\alpha^{2a^*+a_1}) = a_1 = L_2(\beta).$$

Since $L_1(\beta_1)$ can be computed efficiently, we have an algorithm to compute $L_2(\beta)$ efficiently.

6.22 The *ElGamal Cryptosystem* can be implemented in any subgroup $\langle\alpha\rangle$ of a finite multiplicative group $(G, \cdot)$, as follows: Let $\beta \in \langle\alpha\rangle$ and define (α, β) to be the public key. The plaintext space is $\mathcal{P} = \langle\alpha\rangle$, and the encryption operation is $e_K(x) = (y_1, y_2) = (\alpha^k, x \cdot \beta^k)$, where k is random.

Here we show that distinguishing ElGamal encryptions of two plaintexts can be Turing reduced to Decision Diffie-Hellman, and vice versa.

(a) Assume that ORACLEDDH is an oracle that solves Decision Diffie-Hellman in $(G, \cdot)$. Prove that ORACLEDDH can be used as a subroutine in an algorithm that distinguishes ElGamal encryptions of two given plaintexts, say x_1 and x_2. (That is, given $x_1, x_2 \in \mathcal{P}$, and given a ciphertext (y_1, y_2) which is an encryption of x_i for some $i \in \{1, 2\}$, the distinguishing algorithm will determine if $i = 1$ or $i = 2$.)

Answer: For $i = 1, 2$, compute $u_i = y_2(x_i)^{-1}$. If

$$\text{ORACLEDDH}(\alpha, \beta, y_1) = u_i,$$

then (y_1, y_2) is an encryption of x_i.

(b) Assume that ORACLEDISTINGUISH is an oracle that distinguishes ElGamal encryptions of any two given plaintexts x_1 and x_2, for any *ElGamal Cryptosystem* implemented in the group $(G, \cdot)$ as described above. Suppose further that ORACLEDISTINGUISH will determine if a ciphertext (y_1, y_2) is not a valid encryption of either of x_1 or x_2. Prove that ORACLEDISTINGUISH can be used as a subroutine in an algorithm that solves Decision Diffie-Hellman in $(G, \cdot)$.

Answer: We are given an instance of Decision Diffie-Hellman, namely, $\alpha, \beta, \gamma, \delta$. Define $y_1 = \gamma$, $y_2 = \delta$, $x_1 = 1$ and $x_2 \neq 1$ (arbitrarily). Call ORACLEDISTINGUISH$(x_1, x_2, (y_1, y_2))$. If the result is $i = 1$, then answer “yes”; otherwise, answer “no”.

7

Signature Schemes

Exercises

7.1 Suppose Alice is using the *ElGamal Signature Scheme* with $p = 31847$, $\alpha = 5$ and $\beta = 25703$. Compute the values of k and a (without solving an instance of the Discrete Logarithm problem), given the signature $(23972, 31396)$ for the message $x = 8990$ and the signature $(23972, 20481)$ for the message $x = 31415$.

Answer: First, we compute

$k = (x_1 - x_2)(\delta_1 - \delta_2)^{-1} \bmod (p-1) = -22425 \times 10915^{-1} \bmod 31846 = 1165.$

To determine a, we will solve the congruence

$$\gamma a \equiv x_1 - k\delta_1 \pmod{p-1}$$

for a. This congruence simplifies to

$$23972a \equiv 23704 \pmod{31846}.$$

We use the method described on page 285 to solve it. We have that $\gcd(23972, 31846) = 2$, and $2 | 23704$, so the congruence is equivalent to

$$11986a \equiv 11852 \pmod{15923}.$$

This congruence has the solution

$$a \equiv 11852 \times 11986^{-1} \pmod{15923} \equiv 7459 \pmod{15923}.$$

Therefore, $a = 7459$ or $a = 7459 + (p-1)/2 = 23382$. By computing $\alpha^{7459} \bmod p = 25703 = \beta$ and $\alpha^{23382} \bmod p = 6144 \neq \beta$, we see that $a = 7459$.

7.2 Suppose I implement the *ElGamal Signature Scheme* with $p = 31847$, $\alpha = 5$ and $\beta = 26379$. Write a computer program which does the following:

(a) Verify the signature $(20679, 11082)$ on the message $x = 20543$.
Answer: $5^{20543} \bmod 31847 = 20688 = 26379^{20679} 20679^{11082} \bmod 31847.$

(b) Determine my private key, a, by solving an instance of the Discrete Logarithm problem.
Answer: $a = \log_5 26379 = 7973.$

(c) Then determine the random value k used in signing the message x, without solving an instance of the Discrete Logarithm problem.
Answer: To determine k, we will solve the congruence

$$k\delta \equiv x - a\gamma \pmod{p-1}$$

for k. This congruence simplifies to

$$11082k \equiv 13618 \pmod{31846}.$$

We use the method described on page 285 to solve it. $\gcd(11082, 31846) = 2$, and $2|13618$, so the congruence is equivalent to

$$5541k \equiv 6809 \pmod{15923}.$$

This congruence has the solution

$$k \equiv 6809 \times 5541^{-1} \pmod{15923} \equiv 3464 \pmod{15923}.$$

Therefore, $k = 3464$ or $k = 7459 + (p-1)/2 = 19387$. By computing $\alpha^{3464} \bmod p = 11168 \neq \gamma$ and $\alpha^{19387} \bmod p = 20679 = \gamma$, we see that $k = 19387$.

7.3 Suppose that Alice is using the *ElGamal Signature Scheme*. In order to save time in generating the random numbers k that are used to sign messages, Alice chooses an initial random value k_0, and then signs the ith message using the value $k_i = k_0 + 2i \bmod p$ (therefore $k_i = k_{i-1} + 2 \bmod p$ for all $i \geq 1$).

(a) Suppose that Bob observes two consecutive signed messages, say $(x_i, \mathsf{sig}(x_i))$ and $(x_{i+1}, \mathsf{sig}(x_{i+1}))$. Describe how Bob can easily compute Alice's secret key, a, given this information, without solving an instance of the Discrete Logarithm problem. (Note that the value of i does not have to be known for the attack to succeed.)

Answer: Note: the k_i's should be defined modulo $p-1$.

We have the following:

$$k_i\delta_1 \equiv x_1 - a\gamma_1 \pmod{p-1} \tag{7.1}$$

$$(k_i+2)\delta_2 \equiv x_2 - a\gamma_2 \pmod{p-1}. \tag{7.2}$$

Multiply (7.1) by δ_2 and multiply (7.2) by δ_1, obtaining the following:

$$k_i\delta_1\delta_2 \equiv x_1\delta_2 - a\gamma_1\delta_2 \pmod{p-1} \tag{7.3}$$

$$(k_i+2)\delta_1\delta_2 \equiv x_2\delta_1 - a\gamma_2\delta_1 \pmod{p-1}. \tag{7.4}$$

Then compute (7.4) $-$ (7.3):

$$a(\gamma_2\delta_1 - \gamma_1\delta_2) \equiv x_2\delta_1 - x_1\delta_2 - 2\delta_1\delta_2 \pmod{p-1}. \tag{7.5}$$

(7.5) is a linear congruence in the unknown a. There are

$$\gcd(\gamma_2\delta_1 - \gamma_1\delta_2, p-1)$$

solutions to (7.5) modulo $p-1$, and they can be found using the method described on page 285. However, there is a unique correct value of a modulo p that satisfies the condition $\alpha^a \equiv \beta \pmod p$. If (7.5) has more than one solution modulo $p-1$, then it will be necessary to verify which of these solutions is the (unique) correct value of a.

(b) Suppose that the parameters of the scheme are $p = 28703$, $\alpha = 5$ and $\beta = 11339$, and the two messages observed by Bob are

$$\begin{array}{ll} x_i = 12000 & \mathsf{sig}(x_i) = (26530, 19862) \\ x_{i+1} = 24567 & \mathsf{sig}(x_{i+1}) = (3081, 7604). \end{array}$$

Find the value of a using the attack you described in part (a).

Answer: Here, the congruence (7.5) is as follows:

$$14396a \equiv 9964 \pmod{28702}.$$

We have that $\gcd(14396, 28702) = 2$, so this congruence is equivalent to

$$7198a \equiv 4982 \pmod{14351}.$$

This congruence has the solution

$$a = 4982 \times 7198^{-1} \bmod 14351 = 5324.$$

Therefore $a = 5324$ or $a = 5324 + 14351 = 19675$. It can be verified that

$$5^{5324} \bmod 28703 = 17364 \neq \beta$$

and

$$5^{19675} \bmod 28703 = 11339 = \beta.$$

Therefore, $a = 19675$.

7.4 (a) Prove that the second method of forgery on the *ElGamal Signature Scheme*, described in Section 7.3, also yields a signature that satisfies the verification condition.

Answer: We assume that $\alpha^x \equiv \beta^\gamma \gamma^\delta \pmod{p}$. Suppose h, i and j are integers, $0 \leq h, i, j \leq p-2$, and $\gcd(h\gamma - j\delta, p-1) = 1$. Define

$$\begin{aligned} \lambda &= \gamma^h \alpha^i \beta^j \bmod p \\ \mu &= \delta\lambda(h\gamma - j\delta)^{-1} \bmod (p-1), \quad \text{and} \\ x' &= \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \bmod (p-1). \end{aligned}$$

We need to show that

$$\beta^\lambda \lambda^\mu \equiv \alpha^{x'} \pmod{p}.$$

We proceed as follows:

$$\begin{aligned}
& \alpha^{x'} \equiv \beta^\lambda \lambda^\mu \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{x'} \equiv \beta^\lambda (\alpha^i \beta^j \gamma^h)^\mu \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{x'-i\mu} \equiv \beta^{\lambda+j\mu} \gamma^{h\mu} \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{(x'-i\mu)\gamma} \equiv \beta^{(\lambda+j\mu)\gamma} \gamma^{h\mu\gamma} \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{(x'-i\mu)\gamma} \equiv \beta^{(\lambda+j\mu)\gamma} \gamma^{(\lambda+j\mu)\delta} \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{(x'-i\mu)\gamma} \equiv (\beta^\gamma \gamma^\delta)^{(\lambda+j\mu)} \pmod{p} \\
\Longleftrightarrow\quad & \alpha^{(x'-i\mu)\gamma} \equiv (\alpha^x)^{(\lambda+j\mu)} \pmod{p} \\
\Longleftrightarrow\quad & (x' - i\mu)\gamma \equiv x(\lambda + j\mu) \pmod{p-1} \\
\Longleftrightarrow\quad & x'\gamma - x\lambda \equiv \mu(xj + i\gamma) \pmod{p-1} \\
\Longleftrightarrow\quad & (h\gamma - j\delta)(x'\gamma - x\lambda) \equiv \lambda\delta(xj + i\gamma) \pmod{p-1} \\
\Longleftrightarrow\quad & x'\gamma(h\gamma - j\delta) \equiv x\lambda(h\gamma - j\delta) + \lambda\delta(xj + i\gamma) \pmod{p-1} \\
\Longleftrightarrow\quad & x'\gamma(h\gamma - j\delta) \equiv \gamma\lambda(i\delta + xh) \pmod{p-1} \\
\Longleftrightarrow\quad & x'(h\gamma - j\delta) \equiv \lambda(i\delta + xh)(h\gamma - j\delta)^{-1} \pmod{p-1}.
\end{aligned}$$

(b) Suppose Alice is using the *ElGamal Signature Scheme* as implemented in Example 7.1: $p = 467$, $\alpha = 2$ and $\beta = 132$. Suppose Alice has signed the

message $x = 100$ with the signature $(29, 51)$. Compute the forged signature that Oscar can then form by using $h = 102$, $i = 45$ and $j = 293$. Check that the resulting signature satisfies the verification condition.
Answer: $x' = 355$, $\lambda = 363$ and $\mu = 401$. Then

$$\alpha^{x'} \bmod p = 2^{355} \bmod 467 = 255$$

and

$$\beta^{\lambda}\lambda^{\mu} \bmod p = 132^{363}363^{401} \bmod 467 = 255,$$

so the signature $(363, 401)$ on the message 355 is verified.

7.5 (a) A signature in the *ElGamal Signature Scheme* or the *DSA* is not allowed to have $\delta = 0$. Show that if a message were signed with a "signature" in which $\delta = 0$, then it would be easy for an adversary to compute the secret key, a.
Answer: If $\delta = 0$ in the *DSA*, then $(\text{SHA-1}(x)+a\gamma)k^{-1} \equiv 0 \pmod q$ and hence $\text{SHA-1}(x) + a\gamma \equiv 0 \pmod q$. Then $a = -\text{SHA-1}(x)\gamma^{-1} \bmod q$.

(b) A signature in the *DSA* is not allowed to have $\gamma = 0$. Show that if a "signature" in which $\gamma = 0$ is known, then the value of k used in that "signature" can be determined. Given that value of k, show that it is now possible to forge a "signature" (with $\gamma = 0$) for any desired message (i.e., a selective forgery can be carried out).
Answer: If $\gamma = 0$ in the *DSA*, then $\delta = \text{SHA-1}(x)k^{-1} \bmod q$ and hence $k = \text{SHA-1}(x)\delta^{-1} \bmod q$.

Now, given an arbitrary message x_0, define

$$\begin{aligned}\gamma_0 &= 0,\\ \delta_0 &= \text{SHA-1}(x_0)k^{-1} \bmod q,\end{aligned}$$

where k was computed above. Then $e_1 = k$, $e_2 = 0$, and

$$(\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = (\alpha^{k} \bmod p) \bmod q = 0 = \gamma_0,$$

so the "signature" is verified.

(c) Evaluate the consequences of allowing a signature in the *ECDSA* to have $r = 0$ or $s = 0$.
Answer: If $s = 0$, then it is possible to compute the secret key, m, in the same way that a was computed in the *DSA* when $\delta = 0$:

$$m = -\text{SHA-1}(x)r^{-1} \bmod q.$$

If $r = 0$, then the situation is similar to when $\gamma = 0$ in the *DSA*. First, it is possible to compute k: $k = \text{SHA-1}(x)s^{-1} \bmod q$. Then a signature on an arbitrary message x_0 can be computed using this k by defining

$$\begin{aligned}r &= 0,\\ s &= \text{SHA-1}(x_0)k^{-1} \bmod q.\end{aligned}$$

Then $i = k$ and $j = 0$, and it is easily seen that the "signature" (r, s) on the message x_0 is verified.

7.6 Here is a variation of the *ElGamal Signature Scheme*. The key is constructed in a similar manner as before: Alice chooses $\alpha \in \mathbb{Z}_p{}^*$ to be a primitive element, $0 \leq a \leq p-2$ where $\gcd(a, p-1) = 1$, and $\beta = \alpha^{a} \bmod p$. The key $K = (\alpha, a, \beta)$, where α and β are the public key and a is the private key. Let $x \in \mathbb{Z}_p$ be a message to be signed. Alice computes the signature $\mathsf{sig}(x) = (\gamma, \delta)$, where

$$\gamma = \alpha^{k} \bmod p$$

and

$$\delta = (x - k\gamma)a^{-1} \bmod (p-1).$$

The only difference from the original *ElGamal Signature Scheme* is in the computation of δ. Answer the following questions concerning this modified scheme.

(a) Describe how a signature (γ, δ) on a message x would be verified using Alice's public key.
Answer: We have $a\delta + k\gamma \equiv x \pmod{p-1}$, so $\alpha^{a\delta}\alpha^{k\gamma} \equiv \alpha^x \pmod{p}$. Therefore $\beta^{\delta}\gamma^{\gamma} \equiv \alpha^x \pmod{p}$; this is the verification condition.

(b) Describe a computational advantage of the modified scheme over the original scheme.
Answer: The value $a^{-1} \bmod (p-1)$ does not depend on k or x, so it can be precomputed once and for all. In the original version of the *ElGamal Signature Scheme*, a new value $k^{-1} \bmod (p-1)$ must be computed every time a new signature is created.

(c) Briefly compare the security of the original and modified scheme.
Answer: Suppose Oscar tries to forge a signature for a message x. If he chooses a value for γ and tries to solve for δ, he has to solve an instance of the Discrete Logarithm problem. If he chooses a value for δ and tries to solve for γ, he has to solve a congruence of the form $\gamma^{\gamma} \equiv y \pmod{p}$ for γ. This problem does not seem to be one whose difficulty has been studied. In particular, it is a different problem than the one that arises when trying to forge an ElGamal signature by first choosing δ and then trying to solve for γ.

7.7 Suppose Alice uses the *DSA* with $q = 101$, $p = 7879$, $\alpha = 170$, $a = 75$ and $\beta = 4567$, as in Example 7.4. Determine Alice's signature on the message $x = 52$ using the random value $k = 49$, and show how the resulting signature is verified.
Answer: Note: You should take SHA-1$(x) = 52$. Then $\gamma = 59$ and $\delta = 79$. To verify the signature, $e_1 = 16$, $e_2 = 57$ and

$$(170^{16} 4567^{57} \bmod 7879) \bmod 101 = 59.$$

7.8 We showed that using the same value k to sign two messages in the *ElGamal Signature Scheme* allows the scheme to be broken (i.e., an adversary can determine the secret key without solving an instance of the Discrete Logarithm problem). Show how similar attacks can be carried out for the *Schnorr Signature Scheme*, the *DSA* and the *ECDSA*.
Answer: In the *DSA*, Suppose that $k_1 = k_2 = k$. Then $\gamma_1 = \gamma_2 = \gamma$, say, and

$$k(\delta_1 - \delta_2) \equiv \text{SHA-1}(x_1) - \text{SHA-1}(x_2) \pmod{q}.$$

this allows k to be determined, provided that $\delta_1 \neq \delta_2$:

$$k = (\text{SHA-1}(x_1) - \text{SHA-1}(x_2))(\delta_1 - \delta_2)^{-1} \bmod q.$$

Once k is determined, a can be computed, as follows:

$$a = (k\delta_1 - \text{SHA-1}(x_1))\gamma^{-1} \bmod q.$$

The situation with *ECDSA* is very similar. First, k can be computed:

$$k = (\text{SHA-1}(x_1) - \text{SHA-1}(x_2))(s_1 - s_2)^{-1} \bmod q;$$

and then m can be found:

$$m = (ks_1 - \text{SHA-1}(x_1))r^{-1} \bmod q.$$

7.9 Suppose that $x_0 \in \{0,1\}^*$ is a bitstring such that $\text{SHA-1}(x_0) = 00\cdots 0$. Therefore, when used in *DSA* or *ECDSA*, we have that $\text{SHA-1}(x_0) \equiv 0 \pmod q$.

Now we turn to the *Schnorr Signature Scheme*. With high probability, we will have $\gamma_1 \neq \gamma_2$ even when $k_1 = k_2 = k$, because the γ's depend on the messages being signed. It turns out that we can solve for a directly:

$$a = (\delta_1 - \delta_2)(\gamma_1 - \gamma_2)^{-1} \bmod q.$$

(a) Show how it is possible to forge a *DSA* signature for the message x_0.

HINT Let $\delta = \gamma$, where γ is chosen appropriately.

Answer: Define $\gamma = \delta = \beta \bmod q$. Then $e_1 = 0$ and $e_1 = 1$, and

$$(\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \beta \bmod q = \gamma,$$

so the signature is verified.

(b) Show how it is possible to forge an *ECDSA* signature for the message x_0.
Answer: Let $B = (u, v)$ and define $r = s = u \bmod q$. Then $i = 0$, $j = 1$ and $iA + jB = B = (u, v)$, so the signature is verified.

7.10 (a) We describe a potential attack against the *DSA*. Suppose that x is given, let $z = (\text{SHA-1}(x))^{-1} \bmod q$, and let $\epsilon = \beta^z \bmod p$. Now suppose it is possible to find $\gamma, \lambda \in \mathbb{Z}_q{}^*$ such that

$$\left((\alpha\,\epsilon^{\gamma})^{\lambda^{-1} \bmod q}\right) \bmod p \bmod q = \gamma.$$

Define $\delta = \lambda\,\text{SHA-1}(x) \bmod q$. Prove that (γ, δ) is a valid signature for x.
Answer: We have that $e_1 = \lambda^{-1} \bmod q$ and $e_2 = \gamma\lambda^{-1}z \bmod q$. Then (in $\mathbb{Z}_p$) we have that

$$\alpha^{e_1}\beta^{e_2} = \alpha^{\lambda^{-1} \bmod q}\beta^{\gamma\lambda^{-1}z \bmod q} = (\alpha\,\epsilon^{\gamma})^{\lambda^{-1} \bmod q},$$

and it is easily veriifed that the signature is valid.

(b) Describe a similar (possible) attack against the *ECDSA*.
Answer: Define $z = (\text{SHA-1}(x))^{-1} \bmod q$, and let $C = zB$. Suppose it is possible to find $r, \lambda \in \mathbb{Z}_q{}^*$ such that

$$\lambda^{-1}(A + rC) = (u, v),$$

where $u \bmod q = r$. Define $s = \lambda\,\text{SHA-1}(x) \bmod q$; then (r, s) is a valid signature for x.

7.11 In a verification of a signature constructed using the *ElGamal Signature Scheme* (or many of its variants), it is necessary to compute a value of the form $\alpha^c\beta^d$. If c and d are random ℓ-bit exponents, then a straightforward use of the SQUARE-AND-MULTIPLY algorithm would require (on average) $\ell/2$ multiplications and ℓ squarings to compute each of α^c and β^d. The purpose of this exercise is to show that the product $\alpha^c\beta^d$ can be computed much more efficiently.

(a) Suppose that c and d are represented in binary, as in Algorithm 5.5. Suppose also that the product $\alpha\beta$ is precomputed. Describe a modification of Algorithm 5.5, in which at most one multiplication is performed in each iteration of the algorithm.
Answer:

```
Algorithm: MODIFIED SQUARE-AND-MULTIPLY($\alpha, \beta, c, d$)

$\gamma \leftarrow \alpha\beta$
$z \leftarrow 1$
for $i \leftarrow \ell - 1$ downto $0$
  do
    $z \leftarrow z^2$
    if $c_i = 1$
      then
        if $d_i = 1$
          then $z \leftarrow z\gamma$
          else $z \leftarrow z\alpha$
      else
        if $d_i = 1$
          then $z \leftarrow z\beta$
return $(z)$
```

(b) Suppose that $c = 26$ and $d = 17$. Show how your algorithm would compute $\alpha^c\beta^d$, i.e., what are the values of the exponents i and j at the end of each iteration of your algorithm (where $z = \alpha^i\beta^j$).
Answer: The binary representations of c and d are (respectively) 11010 and 10001. After iteration $i = 4$, $z = \alpha\beta$. After iteration $i = 3$, $z = \alpha^3\beta^2$. After iteration $i = 2$, $z = \alpha^6\beta^4$. After iteration $i = 1$, $z = \alpha^{13}\beta^8$. After iteration $i = 0$, $z = \alpha^{26}\beta^{17}$.

(c) Exlpain why, on average, this algorithm requires ℓ squarings and $3\ell/4$ multiplications to compute $\alpha^c\beta^d$, if c and d are randomly chosen ℓ-bit integers.
Answer: The number of squarings is exactly ℓ. In any iteration of the algorithm, a multiplication is done if and only if $(c_i, d_i) \neq (0, 0)$. If we estimate that $(c_i, d_i) = (0, 0)$ occurs with probability $1/4$, then, on average, $3\ell/4$ multiplications are performed.

(d) Estimate the average speedup achieved, as compared to using the original SQUARE-AND-MULTIPLY algorithm to compute α^c and β^d separately, assuming that a squaring operation takes roughly the same time as a multiplication operation.
Answer: If we compute α^c and β^d separately and then multiply them together, then, on average, we require $3\ell/2 + 3\ell/2 + 1 = 3\ell + 1$ squarings and/or multiplications.

If we compute $\alpha^c\beta^d$ using the algorithm desribed above, then require $7\ell/4$ squarings and/or multiplications. The ratio of the running times of these two approaches is $(7\ell/4)/(3\ell + 1) \approx 7/12$, so the speedup factor is $5/12$ or 42%.

7.12 Prove that a correctly constructed signature in the *ECDSA* will satisfy the verification condition.
Answer: We have that

$$\begin{aligned} iA + jB &= w\text{SHA-1}(x)A + wrmA \\ &= s^{-1}(\text{SHA-1}(x) + mr)A \\ &= kA. \end{aligned}$$

Therefore, $iA + jB = (u, v)$, where $u \bmod q = r$.

7.13 Let E denote the elliptic curve $y^2 \equiv x^3 + x + 26 \bmod 127$. It can be shown that $\#E = 131$, which is a prime number. Therefore any non-identity element in E is a generator for $(E, +)$. Suppose the *ECDSA* is implemented in E, with $A = (2, 6)$ and $m = 54$.

(a) Compute the public key $B = mA$.
Answer: $B = (24, 44)$.

(b) Compute the signature on a message x if SHA-1$(x) = 10$, when $k = 75$.
Answer: The signature is $(88, 60)$.

(c) Show the computations used to verify the signature constructed in part (b).
Answer: We have $w = 107$, $i = 22$, $j = 115$, $iA + jB = (88, 55) = (u, v)$, and then $r = 88 = u$, so the signature is verified.

7.14 In the *Lamport Signature Scheme*, suppose that two k-tuples, x and $x\,'$, were signed by Alice using the same key. Let ℓ denote the number of coordinates in which x and x' differ, i.e.,

$$\ell = |\{i : x_i \neq x'_i\}|.$$

Show that Oscar can now sign $2^{\,\ell} - 2$ new messages.
Answer: Oscar can sign a message $x\,''$ if and only if $x_i'' \in \{x_i, x_i'\}$ for all i, $1 \leq i \leq k$. There are $k - \ell$ indices i where x_i'' is determined uniquely (namely, for those i such that $x_i = x_i'$); and ℓ indices i where x_i'' can be chosen ot be 0 or 1 arbitrarily. The total number of possibilities for x'' is 2^{ℓ}. But x'' is required to be a new message, so $x\,'' \neq x, x'$. Hence, there are $2^{\ell} - 2$ new messages that can be signed by Oscar.

7.15 Suppose Alice is using the *Chaum-van Antwerpen Signature Scheme* as in Example 7.7. That is, $p = 467$, $\alpha = 4$, $a = 101$ and $\beta = 449$. Suppose Alice is presented with a signature $y = 25$ on the message $x = 157$ and she wishes to prove it is a forgery. Suppose Bob's random numbers are $e_1 = 46$, $e_2 = 123$, $f_1 = 198$ and $f_2 = 11$ in the disavowal protocol. Compute Bob's challenges, c and d, and Alice's responses, C and D, and show that Bob's consistency check will succeed.
Answer: Note: Bob's challenges are c and C, and Alice's responses are d and D.
Here $c = 280$, $d = 193$, $C = 17$ and $D = 21$. Then,

$$(d\alpha^{-e_2})^{f_1} \equiv 137 \equiv (D\alpha^{-f_2})^{e_1} \pmod{p}.$$

7.16 Prove that each equivalence class of keys in the *Pedersen-van Heyst Signature Scheme* contains q^2 keys.
Answer: Suppose that $\gamma = \alpha^{c} \bmod p$ and $\beta = \alpha^{a_0} \bmod p$. Then $\gamma \equiv \alpha^{a_1}\beta^{a_2} \pmod{p}$ if and only if $c \equiv a_1 + a_0 a_2 \pmod{q}$. For any $a_2 \in \mathbb{Z}_q$, we can solve for $a_1 \in \mathbb{Z}_q$ uniquely: $a_1 = c - a_0 a_2 \bmod q$. Hence, given γ, we see that

$$|\{(a_1, a_2) : \gamma \equiv \alpha^{a_1}\beta^{a_2} \pmod{p} :\}| = q.$$

K is comprised of two triples of the form (γ, a_1, a_2), and hence there are $q \times q = q^2$ keys in any equivalence class.

7.17 Suppose Alice is using the *Pedersen-van Heyst Signature Scheme*, where $p = 3467$, $\alpha = 4$, $a_0 = 1567$ and $\beta = 514$ (of course, the value of a_0 is not known to Alice).

(a) Using the fact that $a_0 = 1567$, determine all possible keys

$$K = (\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$$

such that $\mathsf{sig}_K(42) = (1118, 1449)$.

Answer: We have the following:

$$\begin{aligned}\gamma_1 &= \alpha^{a_1+1567a_2} \bmod 3467\\ \gamma_2 &= \alpha^{b_1+1567b_2} \bmod 3467\\ a_1+42b_1 &\equiv 1118 \pmod{1733}\\ a_2+42b_2 &\equiv 1449 \pmod{1733}.\end{aligned}$$

We can use the last two equations to express a_1 and a_2 in terms of b_1 and b_2:

$$\begin{aligned}a_1 &= 1118-42b_1 \bmod 1733\\ a_2 &= 1449-42b_2 \bmod 1733.\end{aligned}$$

Then we can simplify the expression for γ_1:

$$\begin{aligned}\gamma_1 &= \alpha^{1118-42b_1+1567(1449-42b_2)} \bmod 3467\\ &= \alpha^{1471-42(b_1+1567b_2)} \bmod 3467\\ &= \alpha^{1471}(\gamma_2)^{-42} \bmod 3467.\end{aligned}$$

Summarizing, the keys K satisfying the given conditions are all 6-tuples of the form $(\gamma_1,\gamma_2,a_1,a_2,b_1,b_2)$, where

$$\begin{aligned}\gamma_1 &= \alpha^{1471-42(b_1+1567b_2)} \bmod 3467,\\ \gamma_2 &= \alpha^{b_1+1567b_2} \bmod 3467,\\ a_1 &= 1118-42b_1 \bmod 1733,\\ a_2 &= 1449-42b_2 \bmod 1733,\end{aligned}$$

and $b_1,b_2 \in \mathbb{Z}_{1733}$

(b) Suppose that $\mathsf{sig}_K(42)=(1118,1449)$ and $\mathsf{sig}_K(969)=(899,471)$. Without using the fact that $a_0=1567$, determine the value of K (this shows that the scheme is a one-time scheme).

Answer: We have the following equations in the unknowns a_1, a_2, b_1 and b_2:

$$\begin{aligned}a_1+42b_1 &\equiv 1118 \pmod{1733}\\ a_2+42b_2 &\equiv 1449 \pmod{1733}\\ a_1+969b_1 &\equiv 899 \pmod{1733}\\ a_2+969b_2 &\equiv 471 \pmod{1733}.\end{aligned}$$

This system is easily solved, yielding $a_1=1313$, $b_1=1357$, $a_2=753$ and $b_2=1502$. These values determine γ_1 and γ_2:

$$\begin{aligned}\gamma_1 &= \alpha^{a_1}\beta^{a_2} \bmod p = 1235, \quad \text{and}\\ \gamma_2 &= \alpha^{b_1}\beta^{b_2} \bmod p = 2112.\end{aligned}$$

Therefore, $K=(1235,2112,1313,753,1357,1502)$.

7.18 Suppose Alice is using the *Pedersen-van Heyst Signature Scheme* with $p=5087$, $\alpha=25$ and $\beta=1866$. Suppose the key is

$$K=(5065,5076,144,874,1873,2345).$$

Now, suppose Alice finds the signature $(2219, 458)$ has been forged on the message 4785.

(a) Prove that this forgery satisfies the verification condition, so it is a valid signature.

Answer: We have that

$$5065 \times 5076^{4785} \bmod 5087 = 1943 = 25^{2219} 1866^{458} \bmod 5087,$$

so the (forged) signature is verified

(b) Show how Alice will compute the proof of forgery, a_0, given this forged signature.

Answer: Alice's true signature on the message $x = 4785$ is $(917, 1983)$. The proof of forgery is the value

$$(2219 - 917)(1983 - 458)^{-1} \bmod 2543 = 2187;$$

note that $25^{2187} \bmod 5087 = 1866$, so the proof of forgery can be verified.