فصل ۱
مقدمه‌ای بر رمزنگاری

§ ۱.۱ رمزهای جانشینی ساده

وقتی جولیوس سزار نبرد را از پایگاه بالای تپه خود بررسی می‌کرد، پیکی خسته و آشفته به حضور او رسیده و یک کاغذ پوستی حاوی حروف نامفهوم

$$j\ s\ j\ r\ d\ k\ f\ q\ q\ n\ s\ l\ g\ f\ h\ p\ g\ w\ j\ f\ p\ y\ m\ w\ t\ z\ l\ m\ n\ r\ r\ s\ j\ s\ y\ q\ z\ h\ n\ z\ x$$

را به دست او داد. در این هنگام، جولیوس فرمانی برای یک واحد از ارابه‌ران‌ها می‌فرستد تا به جناح چپ رفته و از شکافی لحظه‌ای در آرایش حریف بهره برند. چگونه این رشته حروف به ظاهر تصادفی چنین اطلاعات مهمی را نقل می‌کنند؟ این ترفند ساده را توضیح می‌دهیم. به سادگی به جای هر حرف در پیام، حرفی که در الفبا پنج حرف قبل از آن قرار دارد را قرار می‌دهیم. برای مثال حرف j در متن رمزی تبدیل به حرف e در متن ساده[1] می‌شود. از آن‌جا که بعد از حرف e در الفبا، حروف f، g، h، i، j و ...قرار دارند، به کارگیری این روند برای کل متن رمزی نتیجه می‌دهد

[1] متن ساده متن اصلی است که قابل خواندن است و متن رمزی متن رمز شده است.

```
j s j r d k f q q n s l g f h p g w j f p y m w t z l m n r r n s j s y q z h n z x
e n e m y f a l l i n g b a c k b r e a k t h r o u g h i m m i n e n t l u c i u s
```

خط دوم، متن رمزگشایی شده است، و با شکستن آن به کلمات و علامت‌گذاری‌های مناسب جولیوس پیام

Enemy falling back. Breakthrough imminent. Lucius.

را می‌خواند. یک نکته کوچک می‌ماند که باید مورد توجه قرار گیرد. وقتی جولیوس به حرفی مثل d برخورد می‌کند که در الفبا پنج حرف قبل از آن وجود ندارد چه اتفاقی میفتد؟ پاسخ این است که او باید در انتهای الفبا بپیچد. پس d با y جایگزین می‌شود، زیرا y با حروف z، a، b ، cو d دنبال می‌شود.

این پیچش را می‌توان با قرار دادن حروف الفبا روی یک دایره تصور کرد. پس اگر دایره‌ی الفبای دوم را درون دایره‌ی اول قرار داده و دایره‌ی داخلی به اندازه‌ی پنج حرف بچرخد، همان‌طور که در شکل ۱.۱ نشان داده شده است، می‌توان به سادگی از آرایش حاصل برای رمز کردن و رمزگشایی پیام‌های سزار استفاده کرد. برای رمزگشایی یک حرف کافیست آن را روی چرخ درونی پیدا کرده و حرف متناظر با آن را روی چرخ بیرونی بخوانید. برای رمز کردن یک پیام، عکس این روند را طی کنید: حرف متن را روی چرخ خارجی یافته و حرف متن رمزی را از چرخ درونی بخوانید. و توجه کنید که اگر یک چرخ رمزی بسازیم که چرخ درونی‌اش حرکت می‌کند، دیگر به انتقال با پنج حرف محدود نیستیم. چرخ‌های رمزی از این دست برای قرن‌ها مورد استفاده قرار می‌گرفته است[۲].

هر چند جزئیات مرحله قبل ساختگی است، و بعید است پیامی به یک رومی به زبان انگلیسی نوشته شود، شواهدی وجود دارد که قیصر این روش اولیه‌ی رمزنگاری، که گاهی به یاد او رمز سزار خوانده می‌شود، را به کار می‌برده است. از این رمز گاهی اوقات با نام رمز انتقالی نیز یاد می‌شود، زیرا هر حرف الفبا به چپ یا راست انتقال می‌یابد. ریشه‌ی کلمه‌ی رمزنگاری، روش‌شناسی پنهان‌سازی متن پیام‌ها، از کلمات یونانی *kryptos*، به معنای پنهان سازی، و *graphikos* به معنای نوشتن است.

[۲] یک چرخ رمز با الفبای درهم و با رمز کردن با استفاده از برامدگی‌های مختلف برای اجزای مختلف پیام در قرن ۱۵ توسط لئون باتیستا آلبرتی [۵۸] طراحی شد.

شکل ۱. یک چرخ رمز با جابجاسازی ۵ حرف

گاهی اوقات مطالعه مدرن رمزنگاری را با نام رمز ($cryptology$) یاد می‌کنند.

در رمز قیصر، هر حرف با یک حرف خاص جایگزین می‌شود. هر چند، اگر باب با استفاده از رمز قیصر یک پیام را برای آلیس رمز کند و اجازه دهد پیام به دست اِو بیفتد، اِو برای رمزگشایی پیام زمان کمی نیاز دارد. تنها چیزی که او نیاز دارد این است که همه‌ی ۲۶ انتقال ممکن را بیازماید.

باب می‌تواند با استفاده از یک طرح جایگزینی پیچیده‌تر حمله را سخت‌تر کند. برای مثال، او می‌تواند هر وقوع a را با z و هر وقوع z را با a، هر وقوع b را با y و هر وقوع y را با b و ...جایگزین کند.

این مثالی از یک رمز جانشینی ساده است، یعنی رمزی که هر حرف در آن با یک حرف دیگر(یا نمادی از نوع دیگر) جایگزین می‌شود. رمز قیصر مثالی از یک رمز جانشینی ساده است، اما رمزهای جانشینی زیادی به جز رمز قیصر وجود دارند. در حقیقت، یک رمز جانشینی ساده را می‌توان به عنوان یک قاعده یا یک تابع در نظر گرفت

$$\{a, b, c, d, e, \ldots, x, y, z\} \longrightarrow \{A, B, C, D, E, \ldots, X, Y, Z\}$$

که به هر حرف متن در دامنه یک حرف رمز متفاوت در برد اختصاص می‌دهد. (برای این‌که تشخیص متن ساده از متن رمز را آسان‌تر کنیم، متن ساده را با استفاده از حروف کوچک و متن رمزی را با

استفاده از حروف بزرگ می‌نویسیم.) توجه کنید برای این‌که رمزگشایی کار کند، تابع رمز کردن باید دارای این خاصیت باشد که هیچ دو حرفی در متن ساده به یک حرف در متن رمز نگاشته نشود. یک تابع با این خاصیت را یک به یک یا اینجکتیو می‌نامیم.

یک راه مناسب برای توصیف تابع رمز کردن، ساخت یک جدول است که در آن الفبای متن ساده در سطر بالا و هر حرف رمز در زیر حرف متن متناظر قرار می‌گیرد.

مثال ۱.۱ یک جدول رمز جانشینی ساده در جدول آمده است. الفبای متن رمز(حروف بزرگ در سطر پایین) یک جایگشت تصادفی از ۲۶ حرف الفبا است. برای رمز کردن پیام

Four score and seven years ago,

حروف را به ترتیب به دنبال هم قرار می دهیم، در جدول رمزکردن هر حرف متن ساده را مشاهده کرده و حرف متن رمز متناظر را در زیر آن می‌نویسیم.

f	o	u	r	s	c	o	r	e	a	n	d	s	e	v	e	n	y	e	a	r	s	a	g	o
N	U	R	B	K	S	U	B	V	C	G	Q	K	V	E	V	G	Z	V	C	B	K	C	F	U

متداول است که متن رمز را دربلوک‌های پنج حرفی بنویسیم:

NURBK SUBVC CQKVE VGZVC BKCFU

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	I	S	Q	V	N	F	O	W	A	X	M	T	G	U	H	P	B	K	L	R	E	Y	D	Z	J

جدول ۱. جدول رمز جانشینی ساده

رمزگشایی روندی مشابه است.

j	r	a	x	v	g	n	p	b	z	s	t	l	f	h	q	d	u	c	m	o	e	i	k	w	y
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

جدول ۱. جدول رمزگشایی جانشینی ساده

فرض کنید پیام

$$GVVQG\ VYKCM\ CQQBV\ KKWGF\ SCVKV\ B$$

را دریافت کرده‌ایم و می‌دانیم این پیام با استفاده از جدول رمز شده است. می‌توانیم با یافتن هر حرف متن رمز در سطر دوم جدول و نوشتن حرف متناظر آن از سطر بالایی معکوس رمز کردن را انجام دهیم. هر چند، از آن‌جا که حروف سطر دوم جدول همگی در هم شده است، این روند تا اندازه‌ای ناکارآمد است. بهتر است تا یک جدول رمزگشایی بسازیم که در آن حروف متن رمز در سطر زیرین به ترتیب الفبایی لیست شده و حروف متن متناظر در سطر فوقانی در هم باشند. این کار را در جدول انجام داده‌ایم. با استفاده از این جدول، به راحتی می‌توانیم پیام را رمزگشایی کنیم.

G	V	V	Q	G	V	Y	K	C	M	C	Q	Q	B	V	K	K	W	G	F	S	C	V	K	V	B
n	e	e	d	n	e	w	s	a	l	a	d	d	r	e	s	s	i	n	g	c	a	e	s	e	r

با دسته بندی و علامت گذاری مناسب یک درخواست فوری آشکار می‌شود!

$$Need\ new\ salad\ dressing.\ -Caesar$$

§ ۱.۱.۱ تحلیل رمزهای جانشینی ساده

چند رمز جانشینی ساده متفاوت موجود است؟ می‌توانیم با شمردن مقادیر رمز ممکن برای هر حرف تعداد آن‌ها را بشماریم. ابتدا به حرف a یکی از ۲۶ حرف رمز ممکن $A-Z$ را اختصاص دهیم. پس ۲۶ امکان برای a وجود دارد. سپس، از آن‌جا که نمی‌توانیم به b همان حرفی که به a اختصاص داریم را اختصاص دهیم، می‌توانیم به b یکی از ۲۵ حرف رمزی باقیمانده را اختصاص دهیم. پس

$۲۶ \cdot ۲۵ = ۶۵۰$ راه ممکن برای اختصاص حرف به a و b وجود دارد. حال دوتا از حروف رمزی را استفاده کرده‌ایم، پس می‌توانیم یکی از ۲۴ حرف رمزی را به c اختصاص دهیم. و با ادامه این روند در می‌یابیم تعداد راه‌هایی که می‌توان ۲۶ حرف متن را به ۲۶ حرف رمز اختصاص دهیم برابر است با

$$۲۶ \cdot ۲۵ \cdot ۲۴ \ldots \cdot ۴ \cdot ۳ \cdot ۲ \cdot ۱ = ۲۶! = ۴۰۳۲۹۱۴۶۱۱۲۶۶۰۵۶۳۵۵۸۴۰۰۰۰۰۰.$$

پس بیش از $۱۰^{۲۶}$ رمز جانشانی ساده متفاوت موجود است. هر جدول رمز کردن به عنوان یک کلید شناخته می‌شود.

فرض کنید اِو جلوی یکی از پیام‌های باب را می‌گیرد و تلاش می‌کند با آزمودن هر یک از رمزهای جانشینی ممکن آن‌را رمزگشایی کند. روند رمزگشایی یک پیام بدون دانستن کلید اساسی را تجزیه و تحلیل رمز می‌خوانیم. اگر اِو(یا کامپیوتر او) قادر به بررسی یک میلیون الفبای رمز در هر ثانیه باشد، ممکن است آزمودن همه‌ی آن‌ها بیش از $۱۰^{۱۳}$ سال به طول انجامد. اما عمر جهان حدود $۱۰^{۱۰}$ سال تخمین زده شده است. پس اِو تقریباً شانسی برای رمزگشایی پیام باب ندارد، این بدین معنی است که پیام باب امن است و چیزی برای نگرانی او وجود ندارد! یا دارد؟

زمان درسی مهم در بخش کاربردی علم رمزنگاری فرا رسیده است:

> حریف شما همواره از بهترین استراتژی برای شکست شما استفاده می‌کند، نه از استراتژیی که شما می‌خواهید. پس امنیت یک سیستم رمز به بهترین روش شناخته شده برای شکستن آن بستگی دارد. وقتی روش‌های جدید و بهبود یافته توسعه می‌یابند، سطح امنیت تنها بدتر می‌شود و هرگز بهتر نمی‌شود.

علی‌رغم تعداد زیاد رمزهای جانشینی ساده ممکن، شکستن این سیستم‌ها در حقیقت آسان است، و در واقع روزنامه‌ها و مجلات زیادی آن‌ها را در کنار جدول متقاطع نمایش می‌دهند. دلیل این که اِو به راحتی می‌تواند یک رمز جانشینی ساده را تجزیه و تحلیل کند این است که حروف در زبان انگلیسی(یا هر زبان بشری) تصادفی نیستند. به عنوان مثال، در زبان انگلیسی اغلب پس از حرف q، حرف

u می‌آید. این حقیقت که فراوانی ظهور حروف معینی چون e و t نسبت به حروف دیگر مثل f و c بیشتر است، مفیدتر است. جدول ۳۰۱ حروف را با فراوانی‌های نوعی‌شان در متون انگلیسی فهرست می‌کند.

By decreasing frequency			
E	13.11%	M	2.54%
T	10.47%	U	2.46%
A	8.15%	G	1.99%
O	8.00%	Y	1.98%
N	7.10%	P	1.98%
R	6.83%	W	1.54%
I	6.35%	B	1.44%
S	6.10%	V	0.92%
H	5.26%	K	0.42%
D	3.79%	X	0.17%
L	3.39%	J	0.13%
F	2.92%	Q	0.12%
C	2.76%	Z	0.08%

In alphabetical order			
A	8.15%	N	7.10%
B	1.44%	O	8.00%
C	2.76%	P	1.98%
D	3.79%	Q	0.12%
E	13.11%	R	6.83%
F	2.92%	S	6.10%
G	1.99%	T	10.47%
H	5.26%	U	2.46%
I	6.35%	V	0.92%
J	0.13%	W	1.54%
K	0.42%	X	0.17%
L	3.39%	Y	1.98%
M	2.54%	Z	0.08%

Table 1.3: Frequency of letters in English text

همان‌طور که می‌توانید ببینید، بیشترین فراوانی مربوط به حرف e است و سپس به ترتیب حروف t، a، o و n بیشترین فراوانی را دارند.

پس اگر اِو حروف پیام رمزی باب را شمرده و یک جدول فراوانی بسازد، احتمالاً فراوان‌ترین حرف نمایش دهنده‌ی e است و t، a، o و n در بین دیگر حروف فراوان ظاهر می‌شوند. بدین ترتیب، اِو می‌توان امکان‌های مختلف را آزموده و پس از تعداد معینی آزمون و خطا، پیام باب را رمزگشایی کند.

در ادامه این بخش با رمزگشایی پیام داده شده در جدول نشان می‌دهیم چگونه می‌توان یک رمز جانشینی ساده را تجزیه و تحلیل کرد. البته در اینجا شکست یک رمز جانشینی ساده هدف اصلی ما نیست. نکته‌ی کلیدی ما معرفی ایده‌ی تحلیل آماری است، که ثابت می‌شود در سراسر رمزنگاری کاربردهای زیادی دارد. هر چند به جهت کمال همه‌ی جزئیات را بیان می‌کنیم، خواننده ممکن است مایل باشد این موضوع را به طور سطحی مورد مطالعه قرار دهد.

LOJUM	YLJME	PDYVJ	QXTDV	SVJNL	DMTJZ	WMJGG	YSNDL	UYLEO	SKDVC
GEPJS	MDIPD	NEJSK	DNJTJ	LSKDL	OSVDV	DNGYN	VSGLL	OSCIO	LGOYG
ESNEP	CGYSN	GUJMJ	DGYNK	DPPYX	PJDGG	SVDNT	WMSWS	GYLYS	NGSKJ
CEPYQ	GSGLD	MLPYN	IUSCP	QOYGM	JGCPL	GDWWJ	DMLSL	OJCNY	NYLYD
LJQLO	DLCNL	YPLOJ	TPJDM	NJQLO	JWMSE	JGGJG	XTUOY	EOOJO	DQDMM
YBJQD	LLOJV	LOJTV	YIOLU	JPPES	NGYQJ	MOYVD	GDNJE	MSVDN	EJM

جدول ۱. یک رمز جانشینی ساده برای تحلیل

در متن رمزی ۲۹۸ حرف وجود دارد. اولین گام ایجاد یک جدول فراوانی است که نشان می‌دهد هر حرف متن رمزی چندبار ظاهر شده است.

	J	L	D	G	Y	S	O	N	M	P	E	V	Q	C	T	W	U	K	I	X	Z	B	A	F	R	H
Freq	32	28	27	24	23	22	19	18	17	15	12	12	8	8	7	6	6	5	4	3	1	1	0	0	0	0
%	11	9	9	8	8	7	6	6	6	5	4	4	3	3	2	2	2	2	1	1	0	0	0	0	0	0

Table 1.5: Frequency table for Table 1.4—Ciphertext length: 298

در متن رمزی حرف J با بیشترین فراوانی ظاهر شده است، پس به طور موقت حدس می‌زنیم این حرف متناظر با حرف e در متن خام باشد. حروف L(۲۸ بار) و D(۲۷ بار) بعد از J بیشترین فراوانی را دارند، پس می‌توانیم با استفاده از جدول ۳۰۱ حدس بزنیم این‌ها نمایش دهنده‌ی حروف t و a هستند. هرچند، بعید است فراوانی حروف در یک پیام کوتاه دقیقاً همان درصدها در جدول ۱۰۳ باشد. همه‌ی چیزی که می‌توانیم بگوییم این است که در میان حروف متن رمزی حروف L، D، G، Y و S احتمالا نشان دهنده‌ی حروف متن ساده t، a، o، n و r هستند.

th	he	an	re	er	in	on	at	nd	st	es	en	of	te	ed
۱۶۸	۱۳۲	۹۲	۹۱	۸۸	۸۶	۷۱	۶۸	۶۱	۵۳	۵۲	۵۱	۴۹	۴۶	۴۶

جدول ۱. دو حرفی‌های انگلیسی متداول‌تر

LO	OJ	GY	DN	VD	YL	DL	DM	SN	KD	LY	NG	OY	JD	SK	EP	JG	SV	JM	JQ
۹	۷	۶	۶	۵	۵	۵	۵	۵	۴	۴	۴	۴	۴	۴	۴	۴	۴	۴	۴

جدول ۱. دو حرفی‌های انگلیسی متداول‌تر در متن رمز جدول

راه‌های زیادی برای پیشروی وجود دارند. یک راه جستجوی دوحرفی‌ها است. جدول دوحرفی‌هایی که با بیشترین فراوانی در انگلیسی ظاهر می‌شوند را فهرست می‌کند و جدول دوحرفی‌هایی که بیشترین فراوانی در متن رمزی ما دارند را نمایش می‌دهد. دو حرفی‌های LO و OJ بیشترین فراوانی را دارند. تا به حال حدس زده‌ایم که $J = e$، و بر اساس فراوانی آن حدس می‌زنیم L احتمالا یکی از حروف t، a، o، n یا r را نمایش می‌دهد. از آن‌جا که دوحرفی‌هایی که بیشترین فراوانی را در زبان انگلیسی دارند عبارتند از th و he، تساوی‌های تجربی

$$OJ = he \text{ و } LO = th$$

را داریم.

با نوشتن حروف متن ساده مفروض زیر حروف رمز متناظر، حدس‌های $J = e$، $L = t$ و $O = h$ را در متن رمز جایگزین می‌کنیم.

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
the-- -te-- ----e ----- --e-t ---e- --e-- ----t --t-h -----
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
---e- ----- --e-- --e-e t---t h---- ----- ---tt h---h t-h--
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
----- ----- --e-e ----- ----- -e--- ----- ----- --t-- ----e
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
----- ---t- -t--- ----- -h--- e---t ----e --t-t he--- --t--
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
te-th -t--t --the --e-- -e-th e---- e--e- ---h- -hheh -----
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
--e-- tthe- the-- --ht- e---- ----e -h--- ---e- ----- -e-
```

در این‌جا، فراونی‌های متن ساده را در نظر گرفته و تلاش می‌کنیم تا برخی کلمات انگلیسی رایج را حدس بزنیم. برای مثال، در خط دوم سه بلوک

$$VSGLL\ OSCIO\ LGOYG,$$

$$---tt\ h---h\ t-h--.$$

را می‌بینیم. با در نظر گرفتن قطعه‌ی $th---ht$، می‌توانیم حدس بزنیم که این کلمه، کلمه‌ی $thought$ است، که هم ارزی‌های

$$S = o, \quad C = u, \quad I = g.$$

را معرفی می‌کند. پس نتیجه می‌شود

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
the-- -te-- ----e ----- o-e-t ---e- --e-- -o--t --t-h o---u
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
---eo --g-- --eo- --e-e to--t ho--- ----- -o-tt hough t-h--
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
-o--- u--o- --e-e ----- ----- -e--- o---- --o-o --t-o --o-e
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
u---- -o-t- -t--- g-ou- -h--- e-u-t ----e --tot heu-- --t--
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
te-th -tu-t --the --e-- -e-th e--o- e--e- ---h- -hheh -----
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
--e-- tthe- the-- -ght- e---o ----e -h--- ---e- -o--- -e-
```

حال سه حرف ght در خط آخر را در نظر بگیرید. قبل از آن‌ها باید یک حرف صدادار قرار داشته باشد، و تنها حروف صدادار باقی‌مانده عبارتند از a و i، پس حدس می‌زنیم که $Y = i$. بنابراین حروف $itio$ را در خط سوم پیدا می‌کنیم، و حدس می‌زنیم که با یک n دنبال می‌شود، که نتیجه می‌دهد $N = n$. (هیچ دلیلی وجود ندارد که یک حرف نتواند خودش را نمایش دهد، هر چند این اتفاق اغلب در رمزهای پازلی که در روزنامه‌ها وجود دارد ممنوع است.) حال داریم

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
the-- ite-- --i-e ----- o-ent ---e- --e-- ion-t -it-h o---u
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
---eo --g-- n-eo- -ne-e to--t ho--- -n-in -o-tt hough t-hi-
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
-on-- u-ion --e-e --in- ---i- -e--- o--n- --o-o -itio n-o-e
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
u--i- -o-t- -t-in g-ou- -hi-- e-u-t ----e --tot heuni niti-
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
te-th -tunt i-the --e-- ne-th e--o- e--e- ---hi -hheh -----
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
i-e-- tthe- the-- ight- e---o n-i-e -hi-- --ne- -o--n -e-
```

تاکنون، زوج‌های متن ساده/ متن رمزی زیر را بازسازی کرده‌ایم:

	J	L	D	G	Y	S	O	N	M	P	E	V	Q	C	T	W	U	K	I	X	Z	B	A	F	R	H
	e	t	-	-	i	o	h	n	-	-	-	-	-	u	-	-	-	-	g	-	-	-	-	-	-	-
Freq	32	28	27	24	23	22	19	18	17	15	12	12	8	8	7	6	6	5	4	3	1	1	0	0	0	0

به یاد آورید که رایج‌ترین حروف در زبان انگلیسی(جدول ۳۰۱) عبارتند از:

$$e, t, a, o, n, r, i, s, h.$$

تاکنون به e، t، o، n، i و h مقادیر رمزی نسبت داده‌ایم، پس حدس می‌زنیم که D و Gدو حرف از سه حرف a، r، و s باشند. در خط سوم مشاهده می‌کنیم که بلوک $GYLYSN$ بلوک $-ition$ را

می‌دهد، پس به وضوح G باید s باشد. به طور مشابه، در خط پنجم $LJQLO\ DLCNL$ هم ارز $te-th\ -tunt$ است، پس D باید a باشد، نه r. با جایگذاری این زوج‌های جدید $G=s$ و $D=a$ داریم

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
the-- ite-- -ai-e ---a- o-ent a--e- --ess ionat -it-h o-a-u
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
s--eo -ag-a n-eo- ane-e to-at ho-a- ansin -ostt hough tshis
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
-on-- usion s-e-e asin- a--i- -eass o-an- --o-o sitio nso-e
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
u--i- sosta -t-in g-ou- -his- esu-t sa--e a-tot heuni nitia
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
te-th atunt i-the --ea- ne-th e--o- esses ---hi -hheh a-a--
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
i-e-a tthe- the-- ight- e---o nsi-e -hi-a sane- -o-an -e-
```

حال با بازبینی متن به راحتی می‌توان زوج‌های اضافی را پر کرد. برای مثال، حرف گمشده در قطعه‌ی $atunt\ i-the$ در خط پنجم باید l باشد، که $P=l$ را می‌دهد، و حرف گمشده در قطعه‌ی $-osition$ در خط سوم باید p باشد که نتیجه می‌دهد $W=p$. با جایگذاری این‌ها، قطعه‌ی $e-p-ession$ در خط اول را می‌یابیم که $Z=x$ و $M=r$ را می‌دهد، و قطعه‌ی $-on-lusion$ در خط سوم که نتیجه می‌دهد $E=c$. سپس قطعه‌ی $consi-er$ در خط آخر $Q=d$ را می‌دهد و کلمات آغازین $the-riterclai-e-$ باید عبارت $the\ writer\ claimed$ باشد که منجر به $U=w$ و $V=m$ می‌شود. با این تفاسیر داریم

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
thewr iterc laime d--am oment ar-ex press ionat witch o-amu
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
scleo ragla nceo- ane-e to-at homam ansin mostt hough tshis
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
concl usion swere asin- alli- leass oman- propo sitio nso-e
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
uclid sosta rtlin gwoul dhisr esult sappe artot heuni nitia
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
tedth atunt ilthe -lear nedth eproc esses --whi chheh adarr
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
i-eda tthem the-m ightw ellco nside rhima sanec roman cer
```

حال پر کردن حروف باقی‌مانده، شکستن آن‌ها به حروف، نقطه‌گذاری مناسب و بازیابی متن ساده زیر کاری ساده است:

The writer claimed by a momentary expression, a twitch of a muscle or a glance of an eye, to fathom a mans inmost thoughts. His conclusions were as infallible as so many propositions of Euclid. So startling would his results appear to the uninitiated that until they learned the processes by which he had arrived at them they might well consider him as a necromancer.

§ ۲.۱ بخش‌پذیری و بزرگترین مقسوم‌علیه مشترک

بخش عظیمی از رمزنگاری مدرن بر اساس جبر و نظریه‌ی اعداد پایه‌گذاری شده است. پس پیش از کاوش در رمزنگاری، نیاز داریم تا برخی ابزار مهم را معرفی کنیم. در چهار بخش آتی با توصیف و اثبات نتایجی اساسی از جبر و نظریه اعداد این فرایند را آغاز می‌کنیم. اگر تاکنون نظریه‌ی اعداد را در دوره‌ای دیگر مطالعه کرده باشید، مروری مختصر از این موضوع کافی خواهد بود. اما اگر این موضوع برای شما جدید است، مطالعه‌ی دقیق آن و حل تمرین‌های انتهای فصل لازم است.

در ابتدایی‌ترین سطح، نظریه‌ی اعداد مطالعه‌ی اعداد طبیعی

$$۱, ۲, ۳, ۴, \ldots,$$

یا به طور کلی‌تر، مطالعه‌ی اعداد صحیح

$$\ldots, -۵, -۴, -۳, -۲, -۱, ۰, ۱, ۲, ۳, ۴, ۵, \ldots$$

است. مجموعه‌ی اعداد صحیح با نماد $\mathbb{Z}$ نمایش داده می‌شود. اعداد صحیح می‌توانند جمع، تفریق یا ضرب شوند، و آن‌ها در تمام قوانین معمول حساب صدق می‌کنند(قوانین جابجایی، شرکت‌پذیری، پخش‌پذیری، و غیره.). مجموعه‌ی اعداد صحیح با قوانین جمع و ضرب معمولا۲ مثالی از یک حلقه است. برای دیدن مطالب بیشتری از نظریه‌ی حلقه‌ها به بخش ۱.۱۰.۲ مراجعه کنید.

اگر a و b اعداد صحیح باشند، می‌توانیم آن‌ها را جمع کنیم $a + b$، تفریق کنیم $a - b$، و ضرب کنیم $a.b$. در هر حالت، یک عدد صحیح بدست می‌آوریم. خاصیت بسته بودن ویژگی یک حلقه است.

اما اگر بخواهیم درون اعداد صحیح بمانیم، همیشه قادر به تقسیم یک عدد صحیح بر دیگری نیستیم. برای مثال، نمی‌توانیم ۳ را بر ۲ تقسیم کنیم، زیرا هیچ عدد صحیحی برابر با $\frac{۳}{۲}$ نیست. این امر منجر به مفهوم اساسی بخش‌پذیری می‌شود.

تعریف ۲.۱ فرض کنید a و b اعداد صحیح باشند و $b \neq ۰$. می‌گوییم b، a را عاد می‌کند یا a بر bبخش‌پذیر است، اگر عدد صحیح c موجود باشد که

$$a = bc.$$

برای بیان این‌که b، a را عاد می‌کند می‌نویسیم $b|a$. اگر b، a را عاد نکند می‌نویسیم $b \nmid a$.

مثال ۳.۱ داریم ۸۴۷|۴۸۵۳۳۱، زیرا ۴۸۵۳۳۱ = ۸۴۷ · ۵۷۳. از سوی دیگر، ۲۵۹۹۴۳ ∤ ۳۵۵ زیرا وقتی سعی می‌کنیم ۲۵۹۹۴۳ را بر ۳۵۵ تقسیم کنیم به باقی‌مانده‌ی ۸۳ می‌رسیم. به طور دقیق‌تر، ۲۵۹۹۴۳ = ۳۵۵ · ۷۳۲ + ۸۳، پس ۲۵۹۹۴۳ مضربی کامل از ۳۵۵ نیست.

تذکر ۴.۱ توجه کنید که هر عدد صحیح بر یک بخش‌پذیر است. اعداد صحیحی که بر ۲ بخش‌پذیرند اعداد صحیح زوج هستند، و اعداد صحیحی که توسط ۲ عاد نمی‌شوند اعداد صحیح فرد هستند.

بخش‌پذیری تعدادی خاصیت ابتدایی دارد، بخشی از این خواص را در گزاره‌ی بعد می‌آوریم.

گزاره ۵.۱ فرض کنید a, b, c اعداد صحیح باشند.

الف) اگر $a|b$ و $b|c$، آن‌گاه $a|c$.

ب) اگر $a|b$ و $b|a$، آن‌گاه $a = \pm b$.

ج) اگر $a|b$ و $a|c$، آن‌گاه $a|(b+c)$ و $a|(b-c)$.

اثبات. اثبات را به عنوان تمرین به خواننده واگذار می‌کنیم. به تمرین ۶.۱ مراجعه کنید. □

تعریف ۶.۱ مقسوم علیه مشترک اعداد صحیح a و b، عدد صحیح مثبت d است که هر دوی آن‌ها را عاد می‌کند. بزرگترین مقسوم‌علیه مشترک a و b، همچنان که نامش پیشنهاد می‌کند، بزرگترین عدد صحیح مثبت d است که $d|a$ و $d|b$. بزرگترین مقسوم علیه مشترک a و b را با $gcd(a, b)$ نمایش می‌دهیم. اگر احتمال اشتباه نباشد، گاهی اوقات آن‌را با (a, b) نیز نمایش می‌دهیم. *(اگر a و b هر دو صفر باشند، آن‌گاه $gcd(a, b)$ تعریف نشده است.)*

عجیب است که مفهومی به سادگی بزرگترین مقسوم علیه مشترک کاربردهای زیادی دارد. به زودی خواهیم دید که روشی کارا و سریع برای محاسبه‌ی بزرگترین مقسوم علیه مشترک دو عدد صحیح وجود دارد، حقیقتی که نتایجی وسیع و قدرتمند دارد.

مثال ۷.۱ بزرگترین مقسوم علیه مشترک ۱۲ و ۱۸ برابر است با ۶، زیرا ۶|۱۲ و ۶|۱۸ و عدد بزرگتری با این خاصیت وجود ندارد. به طور مشابه،

$$gcd(۷۴۸, ۲۰۲۴) = ۴۴.$$

یک راه برای بررسی درست بودن این موضوع این است که تمام مقسوم علیه‌های ۷۴۸ و ۲۰۲۴ را فهرست کنیم.

مقسوم علیه‌های ۷۴۸ = {۱, ۲, ۴, ۱۱, ۱۷, ۲۲, ۳۴, ۴۴, ۶۸, ۱۸۷, ۳۷۴, ۷۴۸}

مقسوم علیه‌های ۲۰۲۴ = {۱, ۲, ۴, ۸, ۱۱, ۲۲, ۲۳, ۴۴, ۴۶, ۸۸, ۹۲, ۱۸۴, ۲۵۳, ۵۰۶, ۱۰۱۲, ۲۰۲۴}

با بررسی دو فهرست، می‌بینیم که بزرگترین درایه‌ی مشترک ۴۴ است. حتی از این مثال کوچک نیز واضح است این روش کارا نیست. هر وقت نیاز داریم تا بزرگترین مقسوم علیه مشترک اعداد بزرگ را محاسبه کنیم، باید روشی کاراتر را بیابیم.

کلید یک الگوریتم سریع برای محاسبه‌ی بزرگترین مقسوم علیه مشترک تقسیم با باقیمانده است، که همان روش تقسیم طولانی است که در ابتدایی آموختید. پس اگر a و b اعداد صحیح مثبت باشند و اگر a را بر b تقسیم کنید، یک خارج قسمت q و یک باقیمانده‌ی r بدست خواهید آورد، که باقیمانده از b کوچکتر است. برای مثال،

```
        13 R 9
17 ) 230
     17
      60
      51
       9
```

پس تقسیم ۲۳۰ بر ۱۷، خارج قسمت ۱۳ و باقیمانده ۹ را می‌دهد. این به چه معنی است؟ یعنی ۲۱۳ را می‌توان به شکل

$$۲۳۰ = ۱۷ \cdot ۱۳ + ۹$$

نوشت، که باقیمانده ۹ اکیداً کوچکتر از ۱۷ است.

تعریف ۸.۱ (الگوریتم تقسیم) فرض کنید a و b اعداد صحیح مثبت باشند. در این صورت a تقسیم بر b دارای خارج قسمت q و باقی‌مانده‌ی r است یعنی

$$a = b \cdot q + r \qquad ۰ \leq r < b.$$

مقادیر q و r به طور منحصر به فرد توسط a و b تعیین می‌شوند.

حال فرض کنید می‌خواهیم بزرگترین مقسوم علیه مشترک a و b را بیابیم. ابتدا aرا بر b تقسیم می‌کنیم تا بدست آوریم

$$a = b \cdot q + r, \qquad \circ \leq r < b. \tag{۱.۱}$$

اگر d یک مقسوم علیه مشترک a و b باشد، با توجه به معادله‌ی (۱.۱) به وضوح d یک مقسوم علیه r است. (به قسمت (ج) گزاره‌ی ۵.۱ مراجعه کنید.) به طور مشابه، اگر e یک مقسوم علیه مشترک b و r باشد، آن‌گاه معادله‌ی (۱.۱) نشان می‌دهد e یک مقسوم علیه a است. به عبارت دیگر، مقسوم علیه‌های مشترک a و b همان مقسوم علیه‌های مشترک b و r هستند، بنابراین

$$gcd(a, b) = gcd(b, r).$$

با تقسیم b بر r و یافتن یک خارج قسمت و باقیمانده‌ی دیگر، برای مثال

$$b = r \cdot q' + r', \qquad \circ \leq r' < r$$

این روند را ادامه می‌دهیم. در این صورت همان استدلال قبلی نشان می‌دهد

$$gcd(b, r) = gcd(r, r').$$

با ادامه‌ی این روند، باقیمانده کوچک و کوچکتر می‌شود، تا جایی که به باقیمانده صفر می‌رسیم، در این زمان مقدار نهایی $gcd(s, \circ) = s$ برابر بزرگترین مقسوم علیه مشترک a و b است.

با یک مثال این روند را توضیح داده و سپس روش کلی، که از آن با نام الگوریتم اقلیدس یاد می‌شود، را توصیف می‌کنیم.

مثال ۹.۱ با استفاده از الگوریتم اقلیدس، که چیزی به جز تقسیم متوالی بر باقیمانده نیست، $gcd(۲۰۲۴, ۷۴۸)$ را محاسبه می‌کنیم. توجه کنید که چگونه خارج قسمت و باقیمانده در هر خط، a و b جدید در خط بعدی می‌شود:

$$۲۰۲۴ = ۷۴۸ \cdot ۲ + ۵۲۸$$

$$۷۴۸ = ۵۲۸ \cdot ۱ + ۲۲۰$$

$$۵۲۸ = ۲۲۰ \cdot ۲ + ۸۸$$

$$۲۲۰ = ۸۸ \cdot ۲ + ۴۴ \longleftarrow gcd = ۴۴$$

$$۸۸ = ۴۴ \cdot ۲ + ۰$$

قضیه ۱۰.۱ (الگوریتم اقلیدس). فرض کنید a و b اعداد صحیح مثبت باشند و $a \geq b$. الگوریتم آتی $gcd(a, b)$ را در تعدادی متناهی گام محاسبه می‌کند.

قرار بده $r_۰ = a$ و $r_۱ = b$.

قرار بده $i = ۱$.

$r_{i-۱}$ را بر r_i تقسیم کن تا خارج قسمت q_i و باقیمانده‌ی $r_{i+۱}$ را بدست آید،

$$r_{i-۱} = r_i \cdot q_i + r_{i+۱}, \qquad ۰ \leq r_{i+۱} < r_i.$$

اگر باقیمانده‌ی $r_{i+۱} = ۰$، آن‌گاه $r_i = gcd(a, b)$ و الگوریتم خاتمه می‌یابد.

در غیر این‌صورت، $r_{i+۱} > ۰$، پس قرار بده $i = i + ۱$ و به گام سه برو.

گام تقسیم(گام سه) حداکثر

$$۲log_۲(b) + ۱$$

بار اجرا می‌شود.

اثبات. همان‌طور که در شکل ۹.۵ می‌بینیم الگوریتم اقلیدس از دنباله‌ای از تقسیم بر باقیمانده‌ها تشکیل شده است(به یاد آورید که قرار داده‌ایم $r_۰ = a$ و $r_۱ = b$).

مقادیر r_i اکیداً کاهش می‌یابد، و به محض اینکه به صفر برسند الگوریتم خاتمه می‌یابد، که ثابت می‌کند الگوریتم در تعداد متناهی گام خاتمه می‌یابد. به علاوه، در هر تکرار گام سه معادله‌ای به شکل

$$
\begin{array}{ll}
a = b \cdot q_1 + r_2 & \text{with } 0 \le r_2 < b, \\
b = r_2 \cdot q_2 + r_3 & \text{with } 0 \le r_3 < r_2, \\
r_2 = r_3 \cdot q_3 + r_4 & \text{with } 0 \le r_4 < r_3, \\
r_3 = r_4 \cdot q_4 + r_5 & \text{with } 0 \le r_5 < r_4, \\
\vdots \qquad \vdots & \qquad \vdots \\
r_{t-2} = r_{t-1} \cdot q_{t-1} + r_t & \text{with } 0 \le r_t < r_{t-1}, \\
r_{t-1} = r_t \cdot q_t & \\
\multicolumn{2}{c}{\text{Then } r_t = \gcd(a, b).}
\end{array}
$$

شکل ۱. الگوریتم اقلیدسی گام به گام

$$r_{i-۱} = r_i \cdot q_i + r_{i+۱}$$

داریم. این معادله ایجاب می‌کند هر مقسوم‌علیه مشترک $r_{i-۱}$ و r_i یک مقسوم‌علیه $r_{i+۱}$ نیز هست، و به طور مشابه ایجاب می‌کند هر مقسوم‌علیه مشترک r_i و $r_{i+۱}$ نیز یک مقسوم‌علیه $r_{i-۱}$ است. بنابراین

$$gcd(r_{i-۱}, r_i) = gcd(r_i, r_{i+۱}) \quad \forall i = ۱, ۲, ۳, \ldots. \tag{۲.۱}$$

هرچند، همان‌طور که پیش‌تر اشاره شد، در نهایت به یک r_i می‌رسیم که صفر است، برای مثال $r_{t+۱} = ۰$. در این صورت $r_{t-۱} = r_t \cdot q_t$. پس

$$gcd(r_{t-۱}, r_t) = gcd(r_t \cdot q_t, r_t) = r_t.$$

اما معادله‌ی (۲.۱) می‌گوید این برابر $gcd(r_۰, r_۱)$ یعنی $gcd(a, b)$ است، که ثابت می‌کند آخرین باقیمانده در الگوریتم اقلیدس برابر بزرگترین مقسوم‌علیه مشترک a و b است.

تنها تخمین کارایی الگوریتم باقی می‌ماند. پیش‌تر اشاره کردیم از آنجا که مقادیر r_i اکیداً کاهش می‌یابند، الگوریتم خاتمه می‌یابد، و در حقیقت از آنجا که $r_۱ = b$، الگوریتم حتماً در حداکثر b گام خاتمه می‌یابد. هر چند این کران بالا از حقیقت دور است. ادعا می‌کنیم پس از هر دو تکرار گام سه، مقدار r_i حداقل نصف می‌شود. به عبارت دیگر:

ادعا: برای هر $i = ۰, ۱, ۲, \ldots$، $r_{i+۲} < \frac{۱}{۲} r_i$.

با در نظر گرفتن دو حالت، ادعای خود را ثابت می‌کنیم.

حالت اول: $r_{i+۱} \leq \frac{۱}{۲} r_i$

می‌دانیم که مقادیر r_i اکیداً کاهش می‌یابند، بنابراین

$$r_{i+۲} < r_{i+۱} \leq \tfrac{۱}{۲} r_i.$$

حالت دوم: $r_{i+۱} > \frac{۱}{۲} r_i$

در نظر بگیرید وقتی r_i را بر $r_{i+۱}$ تقسیم می‌کنیم چه اتفاقی می‌افتد. مقدار $r_{i+۱}$ به اندازه‌ای بزرگ است که بدست می‌آوریم

$$r_i = r_{i+۱} \cdot ۱ + r_{i+۲}, \quad r_{i+۲} = r_i - r_{i+۱} < r_i - \tfrac{۱}{۲} r_i = \tfrac{۱}{۲} r_i.$$

حال ادعای خود را ثابت کرده‌ایم که برای هر i، $r_{i+۲} < \frac{۱}{۲} r_i$ با استفاده‌ی متناوب از این نامساوی در می‌یابیم که

$$r_{۲k+۱} < \tfrac{۱}{۲} r_{۲k-۱} < \tfrac{۱}{۴} r_{۲k-۳} < \tfrac{۱}{۸} r_{۲k-۵} < \tfrac{۱}{۱۶} r_{۲k-۷} < \ldots < \tfrac{۱}{۲^k} r_۱ = \tfrac{۱}{۲^k} b.$$

پس اگر $۲^k \geq b$، آن‌گاه $r_{۲k+۱} < ۱$، که نتیجه می‌دهد $r_{۲k+۱}$ برابر صفر است و الگوریتم خاتمه می‌یابد. در مفهوم شکل ۹.۵، مقدار $r_{t+۱}$ صفر است، پس داریم $t + ۱ \leq ۲k + ۱$، و لذا $t \leq ۲k$. به علاوه، دقیقاً t تقسیم در شکل وجود دارد، پس الگوریتم اقلیدس در حداکثر $۲k$ تکرار خاتمه می‌یابد. کوچکترین k را انتخاب کنید، پس $۲^k \geq b < ۲^{k-۱}$. لذا

$$\text{تعداد تکرارها} \geq ۲k = ۲(k-۱) + ۲ > ۲log_۲(b) + ۲$$

که اثبات قضیه را کامل می‌کند. □

تذکر ۱۱.۱ ثابت کردیم که الگوریتم اقلیدس که برای a و b با $a \geq b$ به کار می‌رود برای محاسبه‌ی $gcd(a, b)$، به بیش از $۲log_۲(b) + ۱$ تکرار نیاز ندارد. این تقریب می‌تواند تاحدودی بهبود یابد. ثابت

شده است که الگوریتم اقلیدس بیش از $۱/۴۵log_۲(b) + ۱/۶۸$ تکرار ندارد، و تعداد متوسط تکرارها برای a و b تصادفی تقریباً $۰/۸۵log_۲(b) + ۰/۱۴$ است. *(به [۶۱] مراجعه کنید.)*

تذکر ۱۲.۱ یک راه برای محاسبه‌ی خارج قسمت‌ها و باقیمانده‌ها استفاده از تقسیم طولانی است، همان‌طور که در صفحه ؟؟ انجام دادیم. می‌توانید با استفاده از یک ماشین حساب ساده روند را تسریع کنید. اولین گام این است که در ماشین حسابتان a را بر b تقسیم کنید، که عددی حقیقی می‌دهد. قسمتی که بعد از اعشار آمده را دور بریزید تا خارج قسمت q را بیابید. سپس باقیمانده r را می‌توان از رابطه‌ی

$$r = a - b \cdot q$$

محاسبه کرد. برای مثال، فرض کنید $a = ۲۳۸۷۱۸۷$ و $b = ۲۷۵۷۳$. در این صورت $a/b \approx ۸۶/۵۷۶۹۷۷۴۸$، پس $q = ۸۶$ و

$$r = a - b \cdot q = ۲۳۸۷۱۸۷ - ۲۷۵۷۳ \cdot ۸۶ = ۱۵۹۰۹.$$

اگر تنها به باقیمانده احتیاج دارید، می‌توانید در عوض قسمت اعشاری a/b *(که گاهی اوقات بخش کسری خوانده می‌شود)* را بردارید و آن‌را در b ضرب کنید. در مثال ما قسمت اعشاری $a/b \approx ۸۶/۵۷۶۹۷۷۴۸$ برابر است با $۰/۵۷۶۹۷۷۴۸$، و ضرب در $b = ۲۷۵۷۳$ نتیجه می‌دهد

$$۲۷۵۷۳ \cdot ۰/۵۷۶۹۷۷۴۸ = ۱۵۹۰۹/۰۰۰۰۵۶۰۴.$$

گرد کردن این عدد $r = ۱۵۹۰۹$ را می‌دهد.

پس از اجرای الگوریتم اقلیدس روی دو عدد، می‌توانیم مسیر را برگشته و فرمولی فوق العاده جالب را بدست آوریم. پیش از اینکه حکم کلی را بیان کنیم، با یک مثال توضیح می‌دهیم.

مثال ۱۳.۱ به یاد آورید که در مثال ۹.۱ به صورت زیر از الگوریتم اقلیدس برای محاسبه‌ی $gcd(۲۰۲۴, ۷۴۸)$ استفاده کردیم:

$$۲۰۲۴ = ۷۴۸ \cdot ۲ + ۵۲۸$$

$$۷۴۸ = ۵۲۸ \cdot ۱ + ۲۲۰$$

$$۵۲۸ = ۲۲۰ \cdot ۲ + ۸۸$$

$$۲۲۰ = ۸۸ \cdot ۲ + ۴۴ \longleftarrow gcd = ۴۴$$

$$۸۸ = ۴۴ \cdot ۲ + ۰$$

قرار می‌دهیم $a = ۲۰۲۴$ و $b = ۷۴۸$، پس خط اول می‌گوید

$$۵۲۸ = a - ۲b.$$

این را در خط دوم جای‌گذاری می‌کنیم تا بدست آوریم

$$b = (a - ۲b) \cdot ۱ + ۲۲۰,$$

پس

$$۲۲۰ = -a + ۳b.$$

سپس عبارات $۲۲۰ = -a + ۳b$ و $۸۸ = ۳a - ۸b$ را در خط یکی مانده به آخر جای‌گذاری می‌کنیم تا بدست آوریم

$$-a + ۳b = (۳a - ۸b) \cdot ۲ + ۴۴,$$

پس

$$۴۴ = -۷a + ۱۹b.$$

به عبارت دیگر،

$$-۷ \cdot ۲۰۲۴ + ۱۹ \cdot ۷۴۸ = ۴۴ = gcd(۲۰۲۴, ۷۴۸),$$

پس راهی یافتیم تا $gcd(a, b)$ را به صورت یک ترکیب خطی از a و b با ضرایب صحیح بنویسیم.

در حالت کلی، همیشه می‌توان $gcd(a,b)$ را به عنوان ترکیبی خطی با ضرایب صحیح از a و b نوشت.

قضیه ۱۴.۱ (الگوریتم اقلیدس تعمیم یافته). فرض کنید a و b اعداد صحیح مثبت باشند. در این صورت معادله‌ی

$$au + bv = gcd(a,b)$$

همواره جوابی صحیح دارد. (برای الگوریتمی کارا برای یافتن یک جواب به ۱۲.۱ مراجعه کنید.)

اگر $(u_\circ, v_\circ)$ یک جواب باشد، آن‌گاه هر جواب به شکل

$$v = v_\circ - \frac{a \cdot k}{gcd(a,b)} \text{ و } u = u_\circ + \frac{b \cdot k}{gcd(a,b)}$$

است که در آن $k \in \mathbb{Z}$.

اثبات. شکل ۹.۵، که الگوریتم اقلیدس را گام به گام نشان می‌دهد، را در نظر بگیرید. می‌توانیم معادله‌ی اول را برای $r_۲ = a - b \cdot q_۱$ حل کرده و آن‌را در خط دوم جای‌گذاری کنیم تا بدست آوریم

$$b = (a - b \cdot q_۱) \cdot q_۲ + r_۳,$$

پس

$$r_۳ = -a \cdot q_۲ + b \cdot (۱ + q_۱ q_۲).$$

سپس با جای‌گذاری مقادیر بدست آمده برای $r_۲$ و $r_۳$ در خط سوم بدست می‌آوریم

$$a - b \cdot q_۱ = -a \cdot q_۲ + b \cdot (۱ + q_۱ q_۲) q_۳ + r_۴.$$

با مرتب کردن جملات، بدست می‌آوریم

$$r_۴ = a \cdot (۱ + q_۲ q_۳) - b \cdot (q_۱ + q_۳ + q_۱ q_۲ q_۳).$$

نکته کلیدی این است که $r_۴ = a.u + b.v$ که u و v اعداد صحیح هستند. مهم نیست که عبارات u و v بر حسب $q_۱$، $q_۲$ و $q_۳$ نامرتب است. با ادامه‌ی این روش، در هر مرحله در می‌یابیم r_i مجموع مضربی صحیح از a و مضربی صحیح از b است. در نهایت، به $r_t = a \cdot u + b \cdot v$ می‌رسیم که u و v اعداد صحیح هستند. اما $r_t = gcd(a, b)$ که اثبات قسمت اول قضیه را کامل می‌کند. قسمت دوم را به عنوان تمرین به خواننده واگذار می‌کنیم (تمرین ۱۱.۱). □

یک حالت خاص مهم از الگوریتم توسعه یافته‌ی اقلیدس وقتی است که بزرگترین مقسوم‌علیه مشترک a و b یک است. در این حالت به a و b یک نام خاص می‌دهیم.

تعریف ۱۵.۱ فرض کنید a و b اعداد صحیح باشند. می‌گوییم a و b متباین هستند هرگاه $gcd(a, b) = ۱$.

به طور کلی‌تر، هر معادله‌ی

$$Au + Bv = gcd(A, B)$$

می‌تواند با تقسیم هر دو طرف بر $gcd(A, B)$ به حالت اعداد متباین تبدیل شود. بنابراین

$$\frac{A}{gcd(A, B)}u + \frac{B}{gcd(A, B)}v = ۱,$$

که در آن $a = A/gcd(A, B)$ و $b = B/gcd(A, B)$ متباین هستند و در $au + bv = ۱$ صدق می‌کنند. برای مثال دیدیم که بزرگترین مقسوم علیه مشترک ۲۰۲۴ و ۷۴۸ برابر ۴۴ است و

$$-۷ \cdot ۲۰۲۴ + ۱۹ \cdot ۷۴۸ = ۴۴.$$

با تقسیم دو طرف بر ۴۴، بدست می‌آوریم

$$-۷ \cdot ۴۶ + ۱۹ \cdot ۱۷ = ۱.$$

پس $۲۰۲۴/۴۴ = ۴۶$ و $۷۴۸/۴۴ = ۱۷$ متباین هستند، و $u = -۷$ و $v = ۱۹$ ضرایب یک ترکیب خطی ۴۶ و ۱۷ هستند که برابر یک است.

در مثال ۱۳.۱ نشان دادیم چگونه مقادیر الگوریتم اقلیدس را جای‌گذاری کنیم تا $au + bv = gcd(a, b)$ را حل کنیم. تمرین ۱۲.۱ یک الگوریتم کامپیوتر محور برای محاسبه‌ی u و v را توضیح می‌دهد. حال اگر a و b متباین باشند، نسخه‌ای مفهومی از این روند جای‌گذاری را توضیح می‌دهیم. ابتدا با استفاده از مثال $a =$ ۷۳ و $b =$ ۲۵ توضیح می‌دهیم. از الگوریتم اقلیدس داریم

$$۷۳ = ۲۵ \cdot ۲ + ۲۳$$

$$۲۵ = ۲۳ \cdot ۱ + ۲$$

$$۲۳ = ۲ \cdot ۱۱ + ۱$$

$$۲ = ۱ \cdot ۲ + ۰.$$

با استفاده از دنباله‌های خارج قسمت ۲، ۱، ۱۱ و ۲ به صورت زیر یک جعبه ایجاد می‌کنیم

		2	1	11	2
0	1	*	*	*	*
1	0	*	*	*	*

قانون پر کردن خانه‌های جدول به صورت زیر است:

درایه‌ی جدید = (عدد بالایی)· (عدد سمت چپ)+ (عدد دو خانه سمت چپ)

پس دو * سمت چپ برابرند با

$$۲ \cdot ۱ + ۰ = ۲,$$

$$۲ \cdot ۰ + ۱ = ۱.$$

پس جعبه ما حالا به شکل

در می‌آید. سپس دو *بعدی برابرند با

		2	1	11	2
0	1	2	*	*	*
1	0	1	*	*	*

$$۱ \cdot ۲ + ۱ = ۳,$$

$$۱ \cdot ۱ + ۰ = ۱,$$

و دوتای بعدی عبارتند از

$$۱۱ \cdot ۳ + ۲ = ۳۵,$$

$$۱۱ \cdot ۱ + ۱ = ۱۲,$$

و درایه‌های آخر برابرند با

$$۲ \cdot ۳۵ + ۳ = ۷۳,$$

$$۲ \cdot ۱۲ + ۱ = ۲۵.$$

جدول کامل برابر است با

		2	1	11	2
0	1	2	3	35	73
1	0	1	1	12	25

توجه کنید که ستون آخر a و b هستند. مهم‌تر اینکه در ستون ماقبل آخر مقادیر $-v$ و u را می‌دهند. بنابراین در این مثال در می‌یابیم که $۷۳ \cdot ۱۲ - ۲۵ \cdot ۳۵ = ۱$. الگوریتم کلی در شکل ۳.۳ آمده است.

§ ۳.۱ حساب پیمانه‌ای

ممکن است در مدرسه‌ی ابتدایی با حساب ساعتی مواجه شده باشید، که در آن پس از عدد ۱۲، عدد یک می‌آید. این منجر به معادلات عجیبی چون

In general, if a and b are relatively prime and if $q_1, q_2, \ldots, q_t$ is the sequence of quotients obtained from applying the Euclidean algorithm to a and b as in Figure 1.2 on page 13, then the box has the form

		q_1	q_2	$\ldots$	q_{t-1}	q_t
0	1	P_1	P_2	$\ldots$	P_{t-1}	a
1	0	Q_1	Q_2	$\ldots$	Q_{t-1}	b

The entries in the box are calculated using the initial values

$$P_1 = q_1, \qquad Q_1 = 1, \qquad P_2 = q_2 \cdot P_1 + 1, \qquad Q_2 = q_2 \cdot Q_1,$$

and then, for $i \geq 3$, using the formulas

$$P_i = q_i \cdot P_{i-1} + P_{i-2} \qquad \text{and} \qquad Q_i = q_i \cdot Q_{i-1} + Q_{i-2}.$$

The final four entries in the box satisfy

$$a \cdot Q_{t-1} - b \cdot P_{t-1} = (-1)^t.$$

Multiplying both sides by $(-1)^t$ gives the solution $u = (-1)^t Q_{t-1}$ and $v = (-1)^{t+1} P_{t-1}$ to the equation $au + bv = 1$.

شکل ۱. حل $au + bv = ۱$ با استفاده از الگوریتم اقلیدس

$$۶ + ۹ = ۳, \qquad ۲ - ۳ = ۱۱$$

می‌شود. این‌ها به نظر عجیب می‌آیند اما با استفاده از حساب ساعتی درست هستند، زیرا برای مثال ساعت ۱۱ سه ساعت قبل از ساعت ۲ است. پس در حقیقت کاری که ما انجام می‌دهیم این است که ابتدا $۲ - ۳ = -۱$ را محاسبه کرده و سپس حاصل را با ۱۲ جمع کنیم. به طور مشابه، ۹ ساعت پس از ساعت ۶ ساعت ۳ است، زیرا $۶ + ۹ - ۱۲ = ۳$.

نظریه‌ی همنهشتی‌ها روشی قوی در نظریه‌ی اعداد است که بر پایه‌ی ایده‌ی ساده‌ی حساب ساعتی است.

تعریف ۱۶.۱ فرض کنید $m \geq ۱$ یک عدد صحیح باشد. می‌گوییم اعداد a و b به پیمانه‌ی m همنهشت هستند اگر تفاضل آن‌ها $a - b$ توسط m عاد شود. برای نشان دادن اینکه a و b به پیمانه‌ی m همنهشت هستند، می‌نویسیم

$$a \equiv b \pmod{m}.$$

عدد m را پیمانه می‌خوانیم.

مثال‌های ساعتی ما را می‌توان با استفاده از پیمانه $m = ۱۲$ به صورت همنهشتی نوشت:

$$۶ + ۹ = ۱۵ \equiv ۳ \pmod{۱۲}, \qquad ۲ - ۳ = -۱ \equiv ۱۱ \pmod{۱۲}.$$

مثال ۱۷.۱ از آن‌جا که ۵ تفاضل $۱۷ - ۷$ را عاد می‌کند، داریم

$$۱۷ \equiv ۷ \pmod{۵}.$$

از سوی دیگر ۱۱ تفاضل $۱۹ - ۶$ را عاد نمی‌کند، پس

$$۱۹ \not\equiv ۶ \pmod{۱۱}.$$

توجه کنید اعدادی که در

$$a \equiv ۰ \pmod{m}$$

صدق می‌کنند آن‌هایی هستند که توسط m عاد می‌شوند، یعنی مضارب m.

دلیل مفید بودن همنهشتی‌ها این است که همان‌طور که گزاره‌ی بعد می‌گوید، آن‌ها شبیه تساوی عمل می‌کنند.

گزاره ۱۸.۱ فرض کنید $m \geq ۱$ یک عدد صحیح باشد.

اگر $a_۱ \equiv a_۲ \pmod{m}$ و $b_۱ \equiv b_۲ \pmod{m}$، آن‌گاه

$$a_۱ \pm b_۱ \equiv a_۲ \pm b_۲ \pmod{m}, \qquad a_۱ \cdot b_۱ \equiv a_۲ \cdot b_۲ \pmod{m}.$$

فرض کنید a یک عدد صحیح باشد. در این صورت عدد صحیح b موجود است که $a \cdot b \equiv ۱ \pmod{m}$ اگر و تنها اگر $gcd(a, b) = ۱$.

اگر چنین bی موجود باشد، می‌گوییم b وارون(ضربی) a به پیمانه‌ی m است.(نمی‌گوییم یک وارون ضربی زیرا وارون‌های یک عدد به پیمانه‌ی m با هم برابرند.)

اثبات.

این را به عنوان تمرین به خواننده واگذار می‌کنیم، به تمرین ۱۴.۱ مراجعه کنید.

ابتدا فرض کنید $gcd(a,m) = ۱$. در این صورت با توجه به قضیه‌ی ۱۴.۱، می‌توانیم اعداد صحیح u و v را به نحوی بیابیم که $au + mv = ۱$. این یعنی $au - ۱ = -mv$ توسط m عاد می‌شود، پس طبق تعریف، $au \equiv ۱ \pmod{m}$. به عبارت دیگر می‌توانیم قرار دهیم $b = v$.

حال فرض کنید b وارون a به پیمانه‌ی m باشد، در این صورت $a \cdot b \equiv ۱ \pmod{m}$. این یعنی عدد صحیح m موجود است به نحوی که $ab - ۱ = cm$، پس $gcd(a,m) = ۱$. این نشان می‌دهد a به پیمانه‌ی m وارون دارد اگر و تنها اگر $gcd(a,m) = ۱$.

□

قسمت (ب) گزاره‌ی ۱۸.۱ می‌گوید اگر $gcd(a,m) = ۱$ آن‌گاه a به پیمانه‌ی m دارای وارونی چون b است. این امر این نتیجه عجیب را در پی دارد که کسر $b^{-۱} = ۱/b$ در دنیای اعداد صحیح به پیمانه‌ی m دارای معنی است.

مثال ۱۹.۱ اعداد صحیح $m = ۵$ و $a = ۲$ را در نظر می‌گیریم. به وضوح $gcd(۲,۵) = ۱$، پس ۲ به پیمانه‌ی ۵ وارون دارد. وارون ۲ به پیمانه‌ی ۵، ۳است زیرا $۲ \cdot ۳ \equiv ۱ \pmod{۵}$، پس $۲^{-۱} \equiv ۳ \pmod{۵}$. به طور مشابه $gcd(۴,۱۵) = ۱$ پس $۴^{-۱}$ به پیمانه‌ی ۱۵ موجود است. در حقیقت $۴ \cdot ۴ \equiv ۱ \pmod{۱۵}$ پس در پیمانه‌ی ۱، ۴ وارون خودش است.

ما حتی می‌توانیم در زمانی‌که d نسبت به m اول است، با کسر a/d نیز کار کنیم. برای مثال می‌توانیم ۵/۷ به پیمانه‌ی ۱۱ را محاسبه کنیم، بدین منظور ابتدا می‌بینیم که $۷ \cdot ۸ \equiv ۱ \pmod{۱۱}$، پس $۷^{-۱} \equiv ۸ \pmod{۱۱}$. سپس

$$\frac{۵}{۷} = ۵ \cdot ۷^{-۱} \equiv ۵ \cdot ۸ \equiv ۴۰ \equiv ۷ \pmod{۱۱}.$$

تذکر ۲۰.۱ در مثال قبل، محاسبه‌ی وارون به پیمانه‌ی m با استفاده از روش آزمون و خطا کار آسانی بود. اما وقتی m بزرگ است، محاسبه‌ی a^{-1} به پیمانه‌ی m دشوارتر خواهد بود. توجه کنید با الگوریتم اقلیدس تعمیم یافته نشان دادیم وارون‌ها موجودند(قضیه‌ی ۱۴.۱). به منظور محاسبه‌ی u و v ظاهر شده در معادله‌ی $au + mv = gcd(a, m)$ می‌توانیم همان‌طور که در مثال ۱۳.۱ انجام دادیم به طور مستقیم الگوریتم اقلیدس را به کار بریم، یا روش تا حدودی کاراتر جعبه که در انتهای بخش قبلی آمد را استفاده کنیم، یا می‌توانیم از الگوریتم داده شده در تمرین ۱۲.۱ را به کار بریم. در هر حالت، از آن‌جا که الگوریتم اقلیدس برای محاسبه‌ی $gcd(a, b)$ تنها به $2log_2(b) + 3$ تکرار نیاز دارد، محاسبه‌ی a^{-1} به پیمانه‌ی m تنها مضربی کوچک از $log_2(m)$ گام دارد.

حال توسعه‌ی نظریه‌ی حساب پیمانه‌ای را ادامه می‌دهیم. اگر تقسیم a بر m دارای خارج قسمت q و باقیمانده‌ی r باشد، می‌توان آن‌را به شکل

$$a = m \cdot q + r, \quad 0 \leq r < m$$

نوشت. این نشان می‌دهد r بین 0 و $m - 1$ وجود دارد که $a \equiv r \pmod m$، پس اگر بخواهیم با اعداد صحیح به پیمانه‌ی m کار کنیم کافیست اعداد صحیح $0 \leq r < m$ را استفاده کنیم. این امر منجر به تعریف زیر می‌شود.

تعریف ۲۱.۱ می‌نویسیم

$$\mathbb{Z}/m\mathbb{Z} = \{0, 1, 2, \ldots, m - 1\}$$

و $\mathbb{Z}/m\mathbb{Z}$ را حلقه‌ی اعداد صحیح به پیمانه‌ی m می‌خوانیم. توجه کنید وقتی جمع یا ضرب در $\mathbb{Z}/m\mathbb{Z}$ انجام می‌دهیم، همیشه حاصل را بر m تقسیم کرده و برای بدست آوردن عضوی در $\mathbb{Z}/m\mathbb{Z}$، باقیمانده‌ی تقسیم را در نظر می‌گیریم.

شکل ۴.۳ با ارائه جدول کامل جمع و ضرب به پیمانه‌ی ۵، حلقه‌ی $\mathbb{Z}/5\mathbb{Z}$ را توصیف می‌کند.

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

شکل ۱. جدول‌های جمع و ضرب به پیمانه‌ی ۵

تعریف ۲۲.۱ قسمت *(ب)* گزاره‌ی ۱۸.۱ می‌گوید a به پیمانه‌ی m وارون دارد اگر و تنها اگر $gcd(a,m) = ۱$. اعدادی که وارون دارند یکال خوانده می‌شوند. مجموعه‌ی یکال‌ها را با

$$\begin{aligned}(\mathbb{Z}/m\mathbb{Z})^* &= \{a \in \mathbb{Z}/m\mathbb{Z} : \ gcd(a,m) = ۱\} \\ &= \{a \in \mathbb{Z}/m\mathbb{Z} : \ a \text{ به پیمانه‌ی } m \text{ وارون دارد}\}\end{aligned}$$

نمایش می‌دهیم. مجموعه‌ی $(\mathbb{Z}/m\mathbb{Z})^*$ را گروه یکال‌ها به پیمانه‌ی m می‌خوانیم.

توجه کنید اگر $a_۱$ و $a_۲$ به پیمانه‌ی m یکال باشند، $a_۱a_۲$ نیز هست.(آیا درستی این مطلب را می‌بینید؟) پس وقتی دو یکال را ضرب می‌کنیم یک یکال بدست می‌آوریم. از سوی دیگر اگر دو یکال را جمع کنیم، اغلب یک یکال بدست نمی‌آوریم.

مثال ۲۳.۱ گروه یکال‌ها به پیمانه‌ی ۲۴ عبارتند از

$$(\mathbb{Z}/۲۴\mathbb{Z})^* = \{۱, ۵, ۷, ۱۱, ۱۳, ۱۷, ۱۹, ۲۳\}.$$

جدول ضرب $(\mathbb{Z}/۲۴\mathbb{Z})^*$ در شکل ۵.۱ آمده است.

مثال ۲۴.۱ گروه یکال‌ها به پیمانه‌ی ۷ برابر است با

$$(\mathbb{Z}/۷\mathbb{Z})^* = \{۱, ۲, ۳, ۴, ۵, ۶\},$$

زیرا تمام اعداد بین ۱ و ۶ نسبت به ۷ اول هستند. جدول ضرب $(\mathbb{Z}/۷\mathbb{Z})^*$ در شکل ۵.۱ آمده است.

·	1	5	7	11	13	17	19	23
1	1	5	7	11	13	17	19	23
5	5	1	11	7	17	13	23	19
7	7	11	1	5	19	23	13	17
11	11	7	5	1	23	19	17	13
13	13	17	19	23	1	5	7	11
17	17	13	23	19	5	1	11	7
19	19	23	13	17	7	11	1	5
23	23	19	17	13	11	7	5	1

Unit group modulo 24

·	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Unit group modulo 7

شکل ۱. گروه‌های یکال‌های $(\mathbb{Z}/۲۴\mathbb{Z})^*$ و $(\mathbb{Z}/۷\mathbb{Z})^*$

در تعداد زیادی از سیستم‌های رمزنگاری که مطالعه خواهیم کرد، مهم است بدانیم گروه یکال‌ها به پیمانه‌ی m چند عضو دارد. حضور این مقدار به حدی هست که آن‌را نام‌گذاری کنیم.

تعریف ۲۵.۱ تابع فی اویلر(که گاهی با نام تابع توتینت اویلر شناخته می‌شود) تابع $\phi(m)$ است که به شکل زیر تعریف می‌شود

$$\phi(m) = \#(\mathbb{Z}/m\mathbb{Z})^* = \#\{۰ \leq a < m \ : \ gcd(a, m) = ۱\}.$$

برای مثال ، از مثال‌های ۲۳.۱ و ۲۴.۱ می‌بینیم $\phi(۲۴) = ۸$ و $\phi(۷) = ۶$.

§ ۱.۳.۱ حساب پیمانه‌ای و رمزهای انتقالی

به یاد آورید که رمز قیصر(یا انتقالی) که در بخش ۱.۱ مطالعه شد با انتقال هر حرف الفبا با تعداد متناهی حرف کار می‌کند. با انتساب یک عدد به هر حرف مثل جدول یک رمز انتقالی را به طور

ریاضی توضیح می‌دهیم.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵

Table ۱. حروف به اعداد انتساب

یک رمز انتقالی با k انتقال حرف متن متناظر به عدد p را به حرف متن رمز متناظر با عدد $p+k$ به پیمانه ۲۶ می‌برد. توجه کنید چگونه استفاده از حساب پیمانه‌ای، در این حالت پیمانه ۲۶، می‌تواند توصیف رمز انتقالی را تسهیل کند. مقدار انتقال هم به عنوان کلید رمز کردن و هم کلید رمزگشایی به کار می‌رود. رمز کردن با فرمول

$$\text{حرف رمز} \equiv \text{حرف متن ساده} + \text{کلید خصوصی} \quad (mod\ ۲۶)$$

داده می‌شود و رمزگشایی با انتقال در جهت عکس کار می‌کند،

$$\text{حرف متن ساده} \equiv \text{حرف متن رمز} - \text{کلید خصوصی} \quad (mod\ ۲۶)$$

به طور مختصر، اگر قرار دهیم

$$p = \text{حرف متن خام}, \quad c = \text{حرف متن رمز}, \quad k = \text{کلید رمز},$$

آن‌گاه

$$c \equiv p + k \ \ (mod\ ۲۶),$$

و

$$p \equiv c - k \ \ (mod\ ۲۶).$$

§ ۲.۳.۱ الگوریتم سریع به توان رساندن

در برخی سیستم‌های رمزنگاری که مطالعه خواهیم کرد، برای مثال سیستم‌های رمز RSA و دیفایه هلمن آلیس و باب باید توان بزرگی از یک عدد g به پیمانه‌ی عدد دیگر N را محاسبه کنند، که در آن N چند صد رقمی است. راه ساده محاسبه‌ی g^A تکرار ضرب با g است. بنابراین

$$g_۱ \equiv g \pmod N \quad , g_۲ \equiv g \cdot g_۱ \pmod N \quad , g_۳ \equiv g \cdot g_۲ \pmod N$$
$$, g_۴ \equiv g \cdot g_۳ \pmod N \quad , g_۵ \equiv g \cdot g_۱ \pmod N, \ldots$$

واضح است که $g_A \equiv g^A \pmod N$ اما اگر A بزرگ باشد این الگوریتم شدنی نیست. برای مثال، اگر $A \approx ۲^{۱۰۰۰}$، آن‌گاه الگوریتم بیش از عمر تخمینی زمین طول خواهد کشید! به وضوح برای مفید بودن آن، نیاز داریم تا راهی بهتر برای محاسبه‌ی g^A به پیمانه‌ی N بیابیم.

ایده‌ی کار بدین صورت است که برای تبدیل محاسبه‌ی g^A به مجذور و ضرب‌های پیاپی، از بسط دودویی A استفاده کنیم. با یک مثال این ایده را توضیح داده و سپس فرمول کلی روش را بیان می‌کنیم.

مثال ۲۶.۱ فرض کنید می‌خواهیم $۳^{۲۱۸}$ را به پیمانه‌ی ۱۰۰۰ محاسبه کنیم. اولین گام نوشتن ۲۱۸ بر حسب مجموع توان‌های ۲ است

$$۲۱۸ = ۲ + ۲^۳ + ۲^۴ + ۲^۶ + ۲^۷.$$

در این صورت $۳^{۲۱۸}$ تبدیل می‌شود به

$$۳^{۲۱۸} = ۳^{۲+۲^۳+۲^۴+۲^۶+۲^۷} = ۳^۲ \cdot ۳^{۲^۳} \cdot ۳^{۲^۴} \cdot ۳^{۲^۶} \cdot ۳^{۲^۷}. \qquad (۳.۱)$$

توجه کنید محاسبه‌ی مقادیر دنباله‌ی

$$۳, ۳^۲, ۳^{۲^۲}, ۳^{۲^۳}, ۳^{۲^۴}, \ldots$$

نسبتاً ساده است. زیرا هر عدد در دنباله، مجذور عدد قبلی است. به علاوه، از آن‌جا که تنها به این مقادیر به پیمانه‌ی ۱۰۰۰ احتیاج داریم، نیازی نیست تا بیش از سه رقم را نگه داریم. در جدول ۶.۱ توان‌های ۳ به پیمانه‌ی ۱۰۰۰ را تا $۳^{۲^۷}$ فهرست شده‌اند.

i	0	1	2	3	4	5	6	7
$3^{2^i} \pmod{1000}$	3	9	81	561	721	841	281	961

شکل ۱. توان‌های مربعی پیاپی ۳ به پیمانه۱۰۰۰.

علی‌رغم این‌که عدد $۳^{۱۲۸} = ۳^{۲^۷}$ توانی نسبتاً بزرگ دارد، ایجاد جدول ۶.۱ تنها به ۷ ضرب احتیاج دارد، زیرا هر درایه مجذور درایه‌ی قبلی است. برای دانستن این‌که کدام توان‌های جدول ۶.۱ باید برای محاسبه‌ی $۳^{۲۱۸}$ نیاز است از (۳.۱) استفاده می‌کنیم. بنابراین

$$\begin{aligned} ۳^{۲۱۸} &= ۳^۲ \cdot ۳^{۲^۳} \cdot ۳^{۲^۴} \cdot ۳^{۲^۶} \cdot ۳^{۲^۷} \\ &\equiv ۹ \cdot ۵۶۱ \cdot ۷۲۱ \cdot ۲۸۱ \cdot ۹۶۱ \pmod{۱۰۰۰} \\ &\equiv ۴۸۹ \pmod{۱۰۰۰}. \end{aligned}$$

یادآوری می‌کنیم که در محاسبه‌ی ۹۰۵۶۱ · ۷۲۱ · ۲۸۱ · ۹۶۱، می‌توانیم پس از هر ضرب حاصل را به پیمانه‌ی ۱۰۰۰ ببریم، پس هیچگاه لازم نیست با اعداد بزرگ کار کنیم. مشاهده می‌کنیم که محاسبه‌ی $۳^{۲۱۸}$ به پیمانه‌ی ۱۰۰۰ تنها ۱۱ ضرب انجام می‌شود. که نسبت به رویکرد ساده بسیار کمتر است. و برای توان‌های بزرگتر تفاوت بیشتر خواهد بود.

رویکرد کلی به کار گرفته شده در مثال ۲۶.۱ نام‌های زیادی دارد، مثل الگوریتم سریع توان رسانی و الگوریتم مجذور-و-ضرب. حال الگوریتم را به طور رسمی‌تر معرفی می‌کنیم.

الگوریتم سریع توان رسانی

گام ۱. بسط دودویی A را محاسبه کن

$$A = A_\circ + A_۱ \cdot ۲ + A_۲ \cdot ۲^۲ + A_۳ \cdot ۲^۳ + \ldots + A_r \cdot ۲^r, \quad A_\circ, \ldots, A_r \in \{\circ, ۱\},$$

می‌توانیم فرض کنیم $A_r = ۱$.

گام ۲. برای $\circ \leq i \leq r$، با مجذور پیاپی $g^{۲^i}$ به پیمانه‌ی N را محاسبه کن،

$$\begin{aligned}
a_\circ &\equiv g && (mod\ N)\\
a_۱ &\equiv a_\circ^۲ \equiv g^۲ && (mod\ N)\\
a_۲ &\equiv a_۱^۲ \equiv g^{۲^۲} && (mod\ N)\\
a_۳ &\equiv a_۲^۲ \equiv g^{۲^۳} && (mod\ N)\\
\vdots \quad & \vdots \qquad\qquad && \vdots\\
a_r &\equiv a_{r-۱}^۲ \equiv g^{۲^r} && (mod\ N).
\end{aligned}$$

هر جمله مجذور قبلی است، پس r ضرب نیاز داریم.

گام ۳. با استفاده از فرمول

$$\begin{aligned}
g^A &= g^{A_\circ + A_۱ \cdot ۲ + A_۲ \cdot ۲^۲ + A_۳ \cdot ۲^۳ + \cdots + A_r \cdot ۲^r}\\
&= g^{A_\circ} \cdot (g^۲)^{A_۱} \cdot (g^{۲^۲})^{A_۲} \cdot (g^{۲^۳})^{A_۳} \cdots (g^{۲^r})^{A_r}\\
&\equiv a_\circ^{A_\circ} \cdot a_۱^{A_۱} \cdot a_۲^{A_۲} \cdot a_۳^{A_۳} \cdots a_r^{A_r} \quad (mod\ N). \qquad (۴.۱)
\end{aligned}$$

g^A را به پیمانه‌ی N محاسبه کن. توجه کنید که مقادیر $a_\circ, a_۱, \ldots, a_r$ در گام ۲ محاسبه شدند. بنابراین حاصل‌ضرب (۴.۱) را می‌توان با استفاده از مقادیر a_iهایی که توان آن‌ها A_i برابر یک است و سپس ضرب کردن آن‌ها در هم بدست آورد. این حداکثر r حاصل‌ضرب دیگر نیاز دارد.

زمان اجرا. محاسبه‌ی g^A حداکثر $۲r$ ضرب به پیمانه‌ی N احتیاج دارد. از آن‌جا که $A \geq ۲^r$، می‌بینیم برای محاسبه‌ی g^A حداکثر $۲log_۲(A)$ ضرب به پیمانه‌ی N انجام می‌شود. پس حتی اگر Aبسیار بزرگ باشد، برای مثال $A \approx ۲^{۱۰۰۰}$، برای یک کامپیوتر آسان است تا با انجام تقریباً ۲۰۰۰

ضرب، ۲^A را به پیمانه‌ی N محاسبه کند.

کارایی. راه‌های زیادی برای کاراتر کردن الگوریتم مجذور-و- ضرب وجود دارند، به ویژه الگوریتم‌هایی که حافظه‌ی کمتری احتیاج دارند، برای مثال به تمرین ۲۴.۱ مراجعه کنید.

§ ۴.۱ اعداد اول، یکتایی تجزیه، و میدان‌های متناهی

در بخش ۳.۱ حساب پیمانه‌ای را مطالعه کرده و دیدیم جمع، تفریق و ضرب اعداد صحیح به پیمانه‌ی m معنی‌دار است. هر چند تقسیم ممکن است مشکل‌ساز باشد، زیرا تنها وقتی می‌توانیم بر a تقسیم کنیم که $gcd(a, m) = ۱$. اما توجه کنید در حالتی که عدد صحیح m اول باشد، می‌توانیم بر هر عضو ناصفر از $\mathbb{Z}/m\mathbb{Z}$ تقسیم کنیم. پیش از بازگشت به حلقه‌ی $\mathbb{Z}/p\mathbb{Z}$ که در آن p یک عدد اول است، با بحثی مختصر درباره‌ی اعداد اول آغاز می‌کنیم.

تعریف ۲۷.۱ عدد صحیح p را اول می‌گوییم اگر $p \geq ۲$ و تنها اعداد صحیحی که p را عاد می‌کنند ۱ و p باشند.

برای مثال، اولین ده عدد اول عبارتند از ۲، ۳، ۵، ۷، ۱۱، ۱۳، ۱۷، ۱۹، ۲۳ و ۲۹، در حالی‌که صدهزارمین عدد اول ۱۲۹۹۷۰۹ و یک میلیونمین عدد اول ۱۵۴۸۵۸۶۳ است. بینهایت عدد اول موجودند، حقیقتی که در یونان باستان شناخته شده بود و به عنوان قضیه‌ای در اصول اقلیدس ظاهر می‌شود. (به تمرین ۲۶.۱ مراجعه کنید.)

یک عدد اول p را بر اساس تعداد عواملی که p را عاد می‌کنند تعریف می‌شود. بنابراین گزاره‌ی آتی، که خاصیتی مفید از اعدادی که توسط p عاد می‌شود را بیان می‌کند، واضح نیست و نیاز به اثباتی دقیق دارد. توجه کنید گزاره برای اعداد مرکب برقرار نیست. برای مثال ۶، ۳۰۱۰ را عاد می‌کند، اما ۶ نه ۳ و نه ۱۰ را عاد نمی‌کند.

گزاره ۲۸.۱ فرض کنید p یک عدد اول باشد، و فرض کنید p حاصل ضرب دو عدد صحیح a و b را عاد کند. در این صورت p حداقل یکی از اعداد a یا b را عاد می‌کند.

به طور کلی‌تر، اگر p حاصل‌ضربی از اعداد صحیح را عاد کند، برای مثال

$$p|a_۱a_۲\dots a_n,$$

آن‌گاه p حداقل یکی از اعداد a_iها را عاد می‌کند.

اثبات. فرض کنید $g = gcd(a,p)$. در این صورت $g|p$، پس $g = ۱$ یا $g = p$. اگر $g = p$، آن‌گاه $p|a$(زیرا $g|a$)، و حکم ثابت شده است. در غیر این صورت $g = ۱$ و قضیه‌ی ۱۴.۱ می‌گوید می‌توانیم اعداد صحیح u و v را به نحوی بیابیم که $au + bv = ۱$. با ضرب دو طرف معادله در b بدست می‌آوریم

$$abu + pbv = b. \tag{۵.۱}$$

طبق فرض p حاصل‌ضرب ab را عاد می‌کند، و قطعاً p عدد pbv را عاد می‌کند، پس p هر دو جمله‌ی سمت چپ معادله‌ی (۵.۱) را عاد می‌کند. پس سمت راست آن را عاد می‌کند، که نشان می‌دهد p، b را عاد می‌کند و اثبات گزاره‌ی ۲۸.۱ کامل می‌شود.

برای اثبات حالت کلی، حاصل‌ضرب را به صورت $a_۱(a_۲a_۳\dots a_n)$ نوشته و حکم اول را با $a = a_۱$ و $b = a_۲a_۳\dots a_n$ به کار می‌بریم. اگر $p|a_۱$، حکم ثابت شده است. در غیر این صورت، $p|a_۲a_۳\dots a_n$، در این حالت با نوشتن این عبارت به شکل $p|a_۲(a_۳a_۴\dots a_n)$، حکم اول می‌گوید $p|a_۲$ یا $p|a_۳a_۴\dots a_n$. با ادامه‌ی این روند، در نهایت یک a_i پیدا می‌کنیم که توسط p عاد می‌شود.

□

به عنوان کاربردی از گزاره‌ی ۲۸.۱، ثابت می‌کنیم هر عدد صحیح مثبت در اصل یک تجزیه‌ی یکتا به حاصل‌ضرب اعداد اول دارد.

قضیه ۲۹.۱ (قضیه‌ی اساسی حساب). فرض کنید $a \geq ۲$ یک عدد صحیح باشد. در این صورت می‌توان a را به صورت حاصل‌ضرب اعداد اول

$$a = p_۱^{e_۱}p_۲^{e_۲}p_۳^{e_۳}\dots p_r^{e_r}$$

تجزیه کرد. به علاوه، این تجزیه تا حد ترتیب یکتاست.

اثبات. اثبات این‌که هر $a \geq ۲$ را می‌توان به حاصل‌ضرب اعداد اول تجزیه کرد سخت نیست. وسوسه می‌شویم که فرض کنیم یکتایی تجزیه نیز واضح است. هر چند، این موضوع درست نیست، یکتایی تجزیه، خاصیتی ظریف از اعداد است. با استفاده از فرم کلی گزاره‌ی ۲۸.۱ آن را اثبات می‌کنیم. (برای دیدن مثالی که در آن یکتایی تجزیه برقرار نیست به توصیف $\mathbb{E}$-حوزه در [۱۲۶، فصل ۷] مراجعه کنید.)

فرض کنید a دو تجزیه به حاصل‌ضرب اعداد اول دارد،

$$a = p_۱ p_۲ \dots p_s = q_۱ q_۲ \dots q_t, \tag{۶.۱}$$

که در آن p_iها و q_iها همگی اول، نه لزوماً مجزا، هستند و s لزوماً با t برابر نیست. از آن‌جا که $p_۱|a$، می‌بینیم $p_۱$ حاصل‌ضرب $q_۱ q_۲ q_۳ \dots q_t$ را عاد می‌کند. بنابراین طبق صورت کلی گزاره‌ی ۲۸.۱، در می‌یابیم $p_۱$ یکی از q_i ها را عاد می‌کند. با تغییر ترتیب q_iها در صورت لزوم، می‌توانیم فرض کنیم $p_۱|q_۱$. اما $p_۱$ و $q_۱$ هر دو اول هستند، پس داریم $p_۱ = q_۱$. پس می‌توانیم آن‌ها را از طرفین (۶.۱) حذف کنیم، که نتیجه می‌دهد

$$p_۲ p_۳ \dots p_s = q_۲ q_۳ \dots q_t.$$

با s بار تکرار این روند، در نهایت به معادله‌ای به شکل

$$۱ = q_{t-s} q_{t-s+۱} \dots q_t$$

می‌رسیم. که فوراً نتیجه می‌شود $t = s$ و تجزیه‌ی اصلی a تا حد ترتیب یکتاست.(برای دیدن اثباتی جزئی‌تر از قضیه‌ی اساسی حساب، می‌توانید به هر کتاب مقدماتی نظریه‌ی اعداد مراجعه کنید، برای مثال [۳۳، ۴۷، ۵۳، ۹۰، ۱۰۱، ۱۲۶].) □

تعریف ۳۰.۱ قضیه‌ی اساسی حساب(۲۹.۱) می‌گوید در تجزیه‌ی عدد صحیح مثبت a به عوامل اول، هر عدد اول توانی ویژه دارد. این توان را با $ord_p(a)$ نمایش داده و آن‌را مرتبه‌ی (یا توان) p در a می‌خوانیم. (برای سادگی، برای تمام اعداد اول قرار می‌دهیم $ord_p(۱) = ۰$.)

برای مثال، تجزیه‌ی ۱۷۲۸ برابر است با $۱۷۲۸ = ۲^{۶}.۳^{۳}$، بنابراین $ord_{۲}(۱۷۲۸) = ۶$، $ord_{۳}(۱۷۲۸) =$ ۳ و برای هر عدد اول $p \geq ۵$، $ord_p(۱۷۲۸) = ۰$. با استفاده از نماد ord_p، تجزیه‌ی a را می‌توان به‌طور مختصر به شکل

$$a = \prod_{primes} p^{ord_p(a)}$$

نوشت. توجه کنید که این حاصل‌ضرب معنادار است، زیرا برای همه به جز تعداد متناهی p، $ord_p(a)$ صفر است.

مفید است تا ord_p را به عنوان یک تابع

$$ord_p : \{۱, ۲, ۳, \ldots\} \longrightarrow \{۰, ۱, ۲, ۳, \ldots, \} \qquad (۷.۱)$$

در نظر بگیریم. این تابع خواصی جالب دارد، که برخی از آن‌ها در تمرین ۲۸.۱ توصیف شده اند.

حال مشاهده می‌کنیم اگر p یک عدد اول باشد، آن‌گاه هر عدد ناصفر به پیمانه‌ی p وارون ضربی دارد. این یعنی هنگام حساب به پیمانه‌ی یک عدد اول p، نه تنها می‌توانیم جمع، تفریق و ضرب انجام دهیم، بلکه می‌توانیم بر اعضای ناصفر تقسیم کنیم، درست مثل اعداد حقیقی. این خاصیت اعداد اول به اندازه‌ای مهم است که می‌توانیم آن‌را به طور رسمی به شکل یک گزاره بیان کنیم.

گزاره ۳۱.۱ فرض کنیم p یک عدد اول باشد. در این صورت هر عضو ناصفر a در $\mathbb{Z}/p\mathbb{Z}$ وارون ضربی دارد، یعنی عدد b موجود است که

$$ab \equiv ۱ \pmod p.$$

این مقدار b را با $Mod\ p$ $a^{-۱}$ نمایش می‌دهیم، یا اگر p مشخص باشد تنها می‌نویسیم $a^{-۱}$.

اثبات. این گزاره حالت خاصی از قسمت (ب) گزاره‌ی ۱۸.۱ است، زیرا اگر $a \in \mathbb{Z}/p\mathbb{Z}$ ناصفر باشد، آن‌گاه $gcd(a, p) = ۱$. □

تذکر ۳۲.۱ الگوریتم توسعه یافته‌ی اقلیدس(قضیه‌ی ۱۴.۱) یک روش محاسباتی کارا برای محاسبه‌ی $a^{-۱} \mod p$ ارائه می‌کند. به راحتی معادله‌ی

$$au + pv = ۱$$

را حل می‌کنیم و سپس $u = a^{-۱} \mod p$. برای مشاهده‌ی روشی دیگر برای محاسبه‌ی $a^{-۱} \mod p$ به تذکر ۳۷.۱ مراجعه کنید.

گزاره‌ی ۳۱.۱ را می‌توان به این صورت نیز بیان کرد که اگر p یک عدد اول باشد، آن‌گاه

$$(\mathbb{Z}/p\mathbb{Z})^* = \{۱, ۲, ۳, ۴, \ldots, p-۱\}.$$

به عبارت دیگر، وقتی عضو ۰ را از $\mathbb{Z}/p\mathbb{Z}$ حذف کنیم، اعضای باقیمانده یکال و تحت ضرب بسته هستند.

تعریف ۳۳.۱ اگر p یک عدد اول باشد، آن‌گاه مجموعه‌ی $\mathbb{Z}/p\mathbb{Z}$ اعداد صحیح به پیمانه‌ی p با قوانین جمع، تفریق، ضرب، و تقسیم مثالی از یک میدان است. اگر جبر مجرد را مطالعه کرده باشید(یا به بخش ۱۰.۲ مراجعه کنید)، می‌دانید که یک میدان نامی کلی برای یک حلقه‌ی(جابجایی) است که در آن هر عضو ناصفر یک وارون ضربی دارد. شما برخی میدان‌های دیگر آشنا هستید، برای مثال میدان اعداد حقیقی $\mathbb{R}$، میدان اعداد گویا $\mathbb{Q}$، و میدان اعداد مختلط $\mathbb{C}$.

میدان $\mathbb{Z}/p\mathbb{Z}$ از اعداد صحیح به پیمانه‌ی p تنها تعداد متناهی عضو دارد. این میدانی متناهی است و اغلب با $\mathbb{F}_p$ نمایش می‌دهیم. بنابراین $\mathbb{F}_p$ و $\mathbb{Z}/p\mathbb{Z}$ دو نماد برای یک شی هستند. به طور مشابه، گروه یکال‌های $(\mathbb{Z}/p\mathbb{Z})^*$ را با $\mathbb{F}_p^*$ نمایش می‌دهیم. میدان‌های متناهی اهمیتی اساسی در سراسر رمزنگاری، و در حقیقت کل ریاضیات دارند.

تذکر ۳۴.۱ هر چند $\mathbb{Z}/p\mathbb{Z}$ و $\mathbb{F}_p$ برای نشان دادن یک مفهوم به کار می‌روند، تساوی اعضا در دو مجموعه تا حدی متفاوت نمایش داده می‌شود. برای $a, b \in \mathbb{F}_p$، تساوی a و b با $a = b$ نمایش داده می‌شود، در حالی که برای $a, b \in \mathbb{Z}/p\mathbb{Z}$، تساوی a و b با همنهشتی به پیمانه p نشان داده می‌شود، یعنی $a \equiv b \pmod p$.

§ ۵.۱ توان‌ها و ریشه‌های اولیه در میدان‌های متناهی

کاربرد میدان‌های متناهی در رمزنگاری اغلب شامل به توان رساندن اعضای $\mathbb{F}_p$ به توان‌های بالا است. می‌دانیم چگونه با استفاده از الگوریتم به توان رساندن که در بخش ۲.۳.۱ توصیف شده است، این کار را انجام دهیم. در این بخش توان‌ها در $\mathbb{F}_p$ را از نقطه نظر ریاضی محض، بررسی می‌کنیم، نتیجه‌ای اساسی منسوب به فرما را اثبات می‌کنیم، و خاصیتی مهم از گروه یکال‌های $\mathbb{F}_p^*$ را بیان می‌کنیم.

با یک مثال ساده آغاز می‌کنیم. جدول توان‌های ۱، ۲، ۳ و ... و ۶ را به پیمانه عدد اول ۷ نمایش می‌دهیم.

$$
\begin{array}{cccccc}
۱^۱ \equiv ۱ & ۱^۲ \equiv ۱ & ۱^۳ \equiv ۱ & ۱^۴ \equiv ۱ & ۱^۵ \equiv ۱ & ۱^۶ \equiv ۱ \\
۲^۱ \equiv ۲ & ۲^۲ \equiv ۴ & ۲^۳ \equiv ۱ & ۲^۴ \equiv ۲ & ۲^۵ \equiv ۴ & ۲^۶ \equiv ۱ \\
۳^۱ \equiv ۳ & ۳^۲ \equiv ۲ & ۳^۳ \equiv ۶ & ۳^۴ \equiv ۴ & ۳^۵ \equiv ۵ & ۳^۶ \equiv ۱ \\
۴^۱ \equiv ۴ & ۴^۲ \equiv ۲ & ۴^۳ \equiv ۱ & ۴^۴ \equiv ۴ & ۴^۵ \equiv ۲ & ۴^۶ \equiv ۱ \\
۵^۱ \equiv ۵ & ۵^۲ \equiv ۴ & ۵^۳ \equiv ۶ & ۵^۴ \equiv ۲ & ۵^۵ \equiv ۳ & ۵^۶ \equiv ۱ \\
۶^۱ \equiv ۶ & ۶^۲ \equiv ۱ & ۶^۳ \equiv ۶ & ۶^۴ \equiv ۱ & ۶^۵ \equiv ۶ & ۶^۶ \equiv ۱
\end{array}
$$

جدول ۱. توان‌های اعداد به پیمانه‌ی ۷

الگوهای جذاب قابل مشاهده‌ی کمی در جدول وجود دارند، به ویژه این حقیقت که ستون آخر سمت راست تماماً شامل یک است. می‌توانیم این عبارت را به این صورت بیان کنیم که برای هر $a = ۱, ۲, ۳, \ldots, ۶$،

$$a^۶ \equiv ۱ \pmod{۷}.$$

البته، این موضوع نمی‌تواند برای تمام مقادیر a درست باشد، زیرا اگر a مضربی از ۷ باشد، آن‌گاه تمام توان‌های آن نیز هست، بنابراین در این حالات $a^n \equiv ۰ \pmod{p}$. از سوی دیگر، اگر a توسط ۷ عاد

نشود، آن‌گاه a به پیمانه‌ی ۷ با یکی از مقادیر ۱، ۲، ۳، . . .، ۶ همنهشت است. بنابراین

$$a^{۶} \equiv \begin{cases} ۰ \pmod{۷} & \text{اگر } ۷|a \\ \\ ۱ \pmod{۷} & \text{اگر } ۷\nmid a \end{cases}$$

آزمایش دیگر اعداد اول، پیشنهاد می‌کند که این مثال حقیقتی کلی را بیان می‌کند.

قضیه ۳۵.۱ (قضیه‌ی کوچک فرما). فرض کنید p یک عدد اول و a عددی صحیح باشد. در این صورت

$$a^{p-۱} \equiv \begin{cases} ۰ \pmod{p} & \text{اگر } p|a \\ \\ ۱ \pmod{p} & \text{اگر } p\nmid a \end{cases}$$

اثبات. اثبات‌های زیادی برای قضیه‌ی کوچک فرما وجود دارد. اگر نظریه‌ی گروه‌ها را مطالعه کرده باشید، سریع‌ترین اثبات مشاهده‌ی این حقیقت است که اعضای ناصفر در $\mathbb{F}_p$ گروه $\mathbb{F}_p^*$ از مرتبه‌ی $p-۱$ را تشکیل می‌دهند، بنابراین طبق قضیه‌ی لاگرانژ، مرتبه‌ی هر عضو $\mathbb{F}_p^*$، $p-۱$ را عاد می‌کند. برای کسانی که هنوز درسی در نظریه‌ی گروه‌ها نگذرانده‌اند، اثباتی مستقیم می‌آوریم.

اگر $p|a$، آن‌گاه واضح است که هر توانی از a توسط p عاد می‌شود. پس تنها باید حالتی که $p\nmid a$ را در نظر بگیریم. حال فهرست اعداد

$$a,\ ۲a,\ ۳a,\ \ldots,\ (p-۱)a \pmod{p} \tag{۸.۱}$$

$p-۱$ عدد در این لیست وجود دارد، و ادعا می‌کنیم همه‌ی این اعداد متمایز هستند. برای دیدن این موضوع، دو تا از آن‌ها را انتخاب کنید، برای مثال $ja \pmod{p}$ و $ka \pmod{p}$، و فرض کنید این دو با هم برابرند. این یعنی

$$ja \equiv ka \pmod{p},$$

و لذا

$$(j-k)a \equiv ۰ \pmod{p}.$$

پس p حاصل‌ضرب $(j-k)a$ را عاد می‌کند. پس گزاره‌ی ۲۸.۱ به ما می‌گوید یا p، $j-k$ را عاد می‌کند، یا p، a را عاد می‌کند. هر چند فرض کرده‌ایم که p، a را عاد نمی‌کند، پس نتیجه می‌گیریم p، $j-k$ را عاد می‌کند. اما هر دو عدد j و k بین ۱ و $p-۱$ قرار دارند، پس تفاضل آن‌ها بین $-(p-۲)$ و $p-۲$ قرار دارد. تنها یک عدد بین $-(p-۲)$ و $p-۲$ قرار دارد که توسط p عاد می‌شود، و آن عدد صفر است! این ثابت می‌کند $j-k=۰$، که ثابت می‌کند $ja=ka$. پس نشان داده‌ایم که $p-۱$ عدد در فهرست (۸.۱) همگی متمایزند. این اعداد همچنین ناصفر هستند، زیرا ۱، ۲، ۳، ...، $p-۱$ و a توسط p عاد نمی‌شوند.

برای تکرار رئوس مطالب، نشان داده‌ایم که فهرست اعداد (۸.۱) عبارتست از $p-۱$ عدد متمایز بین ۱ و $p-۱$. اما تنها $p-۱$ عدد متمایز بین ۱ و $p-۱$ وجود دارد، بنابراین فهرست اعداد (۸.۱) باید فهرست ۱، ۲، ...، $p-۱$ در ترتیبی متفاوت باشد.

حال ببینید وقتی همه‌ی اعداد a، $۲a$، $۳a$، ...، $(p-۱)a$ را در هم ضرب کرده و به پیمانه‌ی p کاهش دهیم چه اتفاقی میفتد. این مثل ضرب کردن اعداد ۱، ۲، ۳، ...، $p-۱$ به پیمانه‌ی p است، پس همنهشتی

$$a \cdot ۲a \cdot ۳a \ldots (p-۱)a \equiv ۱ \cdot ۲ \cdot ۳ \ldots (p-۱) \pmod p$$

را به دست می‌آوریم. در سمت چپ $p-۱$ کپی از a داریم. این عامل را بیرون کشیده و از نماد فاکتوریل $(p-۱)! = ۱ \cdot ۲ \ldots (p-۱)$ استفاده می‌کنیم تا بدست آوریم

$$a^{p-۱} \cdot (p-۱)! \equiv (p-۱)! \pmod p.$$

در نهایت، از آن‌جا که $(p-۱)!$ توسط p عاد نمی‌شود اجازه داریم تا $(p-۱)!$ را از طرفین حذف کنیم. (از این حقیقت استفاده کردیم که $\mathbb{F}_p$ میدان است، بنابراین می‌توانیم بر هر عدد ناصفر تقسیم کنیم.) این نتیجه می‌دهد

$$a^{p-۱} \equiv ۱ \pmod p.$$

که اثبات قضیه‌ی کوچک فرما را کامل می‌کند. □

مثال ۳۶.۱ عدد $p = ۱۵۴۸۵۸۶۳$ اول است، بنابراین قضیه‌ی کوچک فرما(قضیه‌ی ۳۵.۱) به ما می‌گوید

$$۲^{۱۵۴۸۵۸۶۲} \equiv ۱ \pmod{۱۵۴۸۵۸۶۳}.$$

بنابراین بدون هیچ محاسبه‌ای، می‌دانیم عدد $۲^{۱۵۴۸۵۸۶۲} - ۱$ که بیش از دو میلیون رقم دارد مضربی از ۱۵۴۸۵۸۶۳ است.

تذکر ۳۷.۱ قضیه‌ی کوچک فرما(قضیه‌ی ۳۵.۱) و الگوریتم به توان رساندن سریع(بخش ۲۰.۳.۱) یک روش کارا برای محاسبه‌ی وارون به پیمانه‌ی p فراهم می‌کند، یعنی

$$a^{-۱} \equiv a^{p-۲} \pmod{p}.$$

این همنهشتی درست است زیرا اگر $a^{p-۲}$ را در a ضرب کنیم، آن‌گاه بنابر قضیه‌ی کوچک فرما حاصل به پیمانه‌ی p برابر یک است. این روشی دیگر برای روش الگوریتم توسعه یافته اقلیدس که در تذکر ۱.۳۲ توضیح داده شد ارائه می‌کند. در عمل، دو الگوریتم یک زمان اجرا دارند.

مثال ۳۸.۱ وارون ۷۸۱۴ به پیمانه‌ی ۱۷۴۴۹ را به دو روش محاسبه می‌کنیم. ابتدا،

$$۷۸۱۴^{-۱} \equiv ۷۸۱۴^{۱۷۴۴۷} \equiv ۱۲۸۴ \pmod{۱۷۴۴۹}.$$

دوم، از الگوریتم توسعه یافته اقلیدس برای حل

$$۷۸۱۴u + ۱۷۴۴۹v = ۱$$

استفاده می‌کنیم. پاسخ $(u, v) = (۱۲۸۴, -۵۷۵)$ است، بنابراین $۷۸۱۴^{-۱} \equiv ۱۲۸۴ \pmod{۱۷۴۴۹}$.

مثال ۳۹.۱ عدد $m = ۱۵۴۸۵۲۰۷$ را در نظر بگیرید. با استفاده از الگوریتم به توان رساندن، محاسبه‌ی

$$۲^{m-۱} = ۲^{۱۵۴۸۵۲۰۶} \equiv ۴۱۳۶۶۸۵ \pmod{۱۵۴۸۵۲۰۷}$$

(با استفاده از کامپیوتر) کار دشواری نیست. مقدار ۱ را بدست نیاوردیم، بنابراین به نظر می‌رسد قضیه‌ی کوچک فرما برای m درست نباشد. این چه می‌گوید؟ اگر m اول بود، قضیه‌ی کوچک فرما می‌گوید باید عدد ۱ را بدست می‌آوردیم. این حقیقت که ۱ را بدست نیاوردیم نشان می‌دهد عدد $m = ۱۵۴۸۵۲۰۶$ اول نیست.

لحظه‌ای بدین مطلب بیندیشید، زیرا حقیقتاً متحیر کننده است. با محاسبه‌ای ساده، به طور قطعی ثابت کردیم m اول نیست، در عین حال هیچ یک از عوامل آن را نمی‌شناسیم!

قضیه‌ی کوچک فرما به ما می‌گوید اگر a عددی اول باشد که توسط p عاد نمی‌شود، آن‌گاه $a^{p-۱} \equiv ۱ \pmod{p}$. هر چند، برای هر مقدار ویژه‌ی a، ممکن است توان‌های کوچک $k \geq ۱$ از a موجود باشند به نحوی که

$$a^k \equiv ۱ \pmod{p}.$$

گزاره ۴۰.۱ فرض کنید p یک عدد اول و a یک عدد صحیح باشد که توسط p عاد نمی‌شود. فرض کنید $a^n \equiv ۱ \pmod{p}$. در این صورت مرتبه‌ی a به پیمانه‌ی p، n را عاد می‌کند. به ویژه، مرتبه‌ی a، $p - ۱$ را عاد می‌کند.

اثبات. فرض کنید k مرتبه‌ی a به پیمانه‌ی p باشد، طبق تعریف $a^k \equiv ۱ \pmod{p}$، و k کوچکترین توان مثبت با این خاصیت است. طبق فرض $a^n \equiv ۱ \pmod{p}$. n را بر k تقسیم می‌کنیم تا بدست آوریم

$$n = kq + r, \quad ۰ \leq r < k.$$

در این صورت،

$$۱ \equiv a^n \equiv a^{kq+r} \equiv (a^k)^q \cdot a^r \equiv ۱^r \cdot a^r \equiv a^r \pmod{p}.$$

اما $r < k$، بنابراین از این‌که k کوچکترین عدد صحیح مثبتی بود که a^k همنهشت با ۱ است، نتیجه می‌گیریم $r = ۰$. بنابراین $n = kq$، پس k، n را عاد می‌کند.

در نهایت، قضیه‌ی کوچک فرما می‌گوید $a^{p-۱} \equiv ۱ \pmod p$، بنابراین k، $p - ۱$ را عاد می‌کند.

□

قضیه‌ی کوچک فرما خاصیتی استثنائی از یکال‌ها(یعنی عناصر ناصفر) در یک میدان متناهی را بیان می‌کند. این بخش را با بحثی مختصر در رابطه با خاصیتی دیگر که هم از نظر تئوری و هم عملی مهم است به پایان می‌بریم.

قضیه ۴۱.۱ *(قضیه‌ی ریشه‌ی اولیه).* فرض کنید p یک عدد اول باشد. در این صورت عضو $g \in \mathbb{F}_p^*$ موجود است که توان‌های آن همه‌ی اعضای $\mathbb{F}_p^*$ را می‌دهد، یعنی،

$$\mathbb{F}_p^* = \{۱, g, g^۲, g^۳, \ldots, g^{p-۲}\}.$$

اعضایی با این خاصیت ریشه‌های اولیه‌ی $\mathbb{F}_p$ یا مولدهای $\mathbb{F}_p^*$ خوانده می‌شوند. آن‌ها اعضایی از $\mathbb{F}_p^*$ هستند که مرتبه‌شان $p - ۱$ است.

اثبات. به فصل ۲۰ مرجع [۱۲۶] یا هر یک از متون [۳۳، ۴۷، ۵۳، ۹۰، ۱۰۱] مراجعه کنید. □

مثال ۴۲.۱ میدان $\mathbb{F}_{۱۱}$ دو ریشه‌ی اولیه دارد، زیرا در $\mathbb{F}_{۱۱}$،

$$۲^۰ = ۱ \quad ۲^۱ = ۲ \quad ۲^۲ = ۴ \quad ۲^۳ = ۸ \quad ۲^۴ = ۵$$
$$۲^۵ = ۱۰ \quad ۲^۶ = ۹ \quad ۲^۷ = ۷ \quad ۲^۸ = ۳ \quad ۲^۹ = ۶.$$

بنابراین همه‌ی ۱۰ عضو ناصفر $\mathbb{F}_{۱۱}$ توسط توان‌های ۲ تولید می‌شوند. از سوی دیگر، ۲ ریشه‌ی اولیه‌ی $\mathbb{F}_{۱۷}$ نیست، زیرا در $\mathbb{F}_{۱۷}$،

$$۲^۰ = ۱ \quad ۲^۱ = ۲ \quad ۲^۲ = ۴ \quad ۲^۳ = ۸ \quad ۲^۴ = ۱۶$$
$$۲^۵ = ۱۵ \quad ۲^۶ = ۱۳ \quad ۲^۷ = ۹ \quad ۲^۸ = ۱,$$

پس پیش از بدست آوردن همه ۱۶ عضو ناصفر به پیمانه‌ی ۱۷ به ۱ رسیدیم. هرچند، ۳ یک ریشه‌ی اولیه برای $\mathbb{F}_{۱۷}$ است، زیرا در $\mathbb{F}_{۱۷}$،

$$۳^{۰} = ۱ \quad ۳^{۱} = ۳ \quad ۳^{۲} = ۹ \quad ۳^{۳} = ۱۰ \quad ۳^{۴} = ۱۳ \quad ۳^{۵} = ۵$$
$$۲^{۶} = ۱۵ \quad ۲^{۷} = ۱۱ \quad ۲^{۸} = ۱۶ \quad ۲^{۹} = ۱۴ \quad ۳^{۱۰} = ۸ \quad ۳^{۱۱} = ۷$$
$$۳^{۱۲} = ۴ \quad ۳^{۱۳} = ۱۲ \quad ۳^{۱۴} = ۲ \quad ۳^{۱۵} = ۶.$$

تذکر ۴۳.۱ اگر p بزرگ باشد، آن‌گاه میدان متناهی $\mathbb{F}_p$ تعدادی ریشه‌ی اولیه دارد. فرمول دقیق می‌گوید که $\mathbb{F}_p$ دقیقاً $\phi(p-۱)$ ریشه‌ی اولیه دارد، که در آن ϕ تابع فی اویلر *(* به صفحه ؟؟؟ مراجعه کنید*)* است. برای مثال، می‌توانید بررسی کنید مجموعه‌ی

$$\{۲, ۳, ۸, ۱۰, ۱۱, ۱۴, ۱۵, ۱۸, ۱۹, ۲۱, ۲۶, ۲۷\}$$

فهرست کامل ریشه‌های اولیه‌ی $\mathbb{F}_{۲۹}$ است. این با $\phi(۲۸) = ۱۲$ مطابقت دارد. به طور کلی‌تر، اگر k، $p-۱$ را عاد کند، آن‌گاه دقیقاً $\phi(k)$ عضو $\mathbb{F}_p^*$ دارای مرتبه‌ی k هستند.

§ ۶.۱ رمزنگاری پیش از عصر کامپیوتر

برای حمله‌ای کوتاه به تاریخ رمزنگاری پیش از کامپیوتر توقف می‌کنیم. امید است این نوشته‌ی مختصر اشتیاق شما برای مطالعه‌ی بیشتر در این موضوع جذاب، که در آن توطئه‌ی سیاسی، ماجرای شهامت، و داستان‌های عشقی با موفقیت‌های تکنیکی نقشی برابر بازی می‌کند برانگیزد.

خاستگاه رمزنگاری در غبار زمان گم شده است، اما احتمالاً پنهان‌نگاری کمی پس از این‌که بشر شروع به استفاده از نوعی ارتباط نوشتاری کرد به وجود آمده است، زیرا تصور می‌شود مفهوم اطلاعات محرمانه باید به آغاز تمدن بازگردد. توصیف‌های ثبت شده‌ی اولیه‌ای از متون رمز در رم باستان وجود دارد، شامل رمز انتقالی جولیوس سزار از بخش ۱۰۱، و قطعاً از آن زمان به بعد، تمدن‌های زیادی از هر دو رمز جانشانی، که در آن هر حرف یا نماد دیگر جایگزین می‌شود، و رمزهای جایگشتی، که در آن ترتیب حروف تغییر می‌کند، استفاده کردند.

اختراع تحلیل رمز، یعنی هنر رمزگشایی بدون دانش قبلی از کلید، جدیدتر است. قدیمی‌ترین متون باقی‌مانده، که شامل مراجعات به نسخه‌های گمشده‌ی قدیمی‌تر است، مربوط به محققان عرب قرون

۱۴ و ۱۵ است. این متون نه تنها رمزهای جایگشتی و جانشانی ساده را توضیح می‌دهند، بلکه اولین نمونه رمز جانشانی هم‌آهنگ است، که رمزیست که در آن یک متن ساده ممکن است با هر یک از چندین حرف رمزی ممکن نمایش داده شود. مهم‌تر، آن‌ها شامل اولین توضیح از روش‌های مختلف تحلیل رمز، شامل استفاده از شمارش فراوانی حروف و احتمال این‌که زوج معینی از حروف در کنار دیگری قرار گیرند.

ضمناً، وقتی اروپا از قرون وسطی خارج شد، دولت‌های سیاسی در ایتالیا و سایر نقاط به ارتباطات امن نیازمند شدند، و هم رمزنگاری و هم تحلیل رمز شروع به توسعه کردند. قدیمی‌ترین رمز جانشانی هم‌آهنگ اروپایی به سال ۱۴۰۱ باز می‌گردد. استفاده از چنین رمزی دانش کنونی تحلیل رمز با استفاده از تحلیل فراوانی را پیشنهاد می‌کند، زیرا تنها دلیل استفاده از یک سیستم هم‌آهنگ سخت کردن چنین تحلیل رمزهایی است.

در قرون ۱۵ و ۱۶ رمزهای معروف به چندالفبایی ایجاد شدند. (در بخش ۴.۲ مثالی از یک رمز چندالفبایی، معروف به رمز *vigenere* را خواهیم دید.) ایده‌ی اصلی این است که هر حرف متن ساده با استفاده از یک رمز جانشانی ساده متفاوت رمز می‌شود. نام «چندالفبایی» به استفاده از الفبای رمز متفاوت، که بسته به نوع خاصی از کلید استفاده می‌شود، اشاره دارد. اگر کلید بزرگ باشد زمانی طولانی صرف می‌شود تا یک حرف رمزی مجدداً مورد استفاده قرار گیرد. در قرن ۱۹ روش‌های آماری برای حل چنین سیستم‌هایی با استفاده از فنونی خاص یا حدس قسمتی از پیام یا کلید توسعه یافتند. یادآوری می‌کنیم که رمزهای ماشینی که نقشی بزرگ در جنگ جهانی دوم بازی کردند ، در اصل، رمزهای چندالفبایی بینهایت پیچیده بودند.

طی قرون ۱۸، ۱۹ و ۲۰ رمزها و کدها هم برای مقاصد سیاسی و هم برای مقاصد نظامی به طور گسترده افزایش یافتند، همچنان‌که روش‌های تحلیل رمز گسترش یافتند، هر چند سطح مهارت از نسلی به نسل دیگر و کشوری به کشور دیگر تفاوتی عمیق داشت. برای مثال، وقتی ایالات متحده در سال ۱۹۱۷ وارد جنگ جهانی اول شد، ارتش ایالات متحده از رمزهایی استفاده می‌کردند که از رمزهای اختراع شده در ایتالیا در سال ۱۶۰۰ ضعیف‌تر بودند، که هر تحلیل‌گر رمز متبحر می‌توانست آن را در

چند ساعت بشکند!

اختراع و گسترش وسیع روش‌های ارتباط جمعی، به ویژه تلگراف، نیاز به رمزهای سیاسی، نظامی و تجاری را آشکار کرد، و داستان‌های جذاب زیادی در خصوص رمزگشایی پیام‌های تلگراف که در تاریخ نقش داشته‌اند وجود دارد. یک مثال، تلگرام زیمرمن رسوا، کافی خواهد بود. با بی‌طرفی ایالات متحده در ۱۹۱۷ در جنگ بین آلمان علیه فرانسه و انگلیس، آلمان‌ها مصمم شدند که بهترین امیدشان برای پیروزی تنگ کردن محاصره‌ی انگلیس با آغاز جنگ زیردریایی بی‌قید در اقیانوس اطلس است. این سیاست، که به معنی قایق‌های شناور از کشورهای بی‌طرف بود، احتمالاً پای ایالات متحده را به جنگ می‌گشود، بنابراین آلمان تصمیم گرفت به مکزیک پیشنهاد اتحاد دهد. در عوض این‌که مکزیک به ایالات متحده حمله کند تا او را از جنگ اروپا منحرف کند، آلمان پیشنهاد کرد در انتهای جنگ، بسیاری از تگزاس کنونی، نیو مکزیک و آریزونا به مکزیک داده شود. سرویس امنیتی انگلیس جلوی این ارتباط را گرفت، و علیرغم اینکه با یکی از امن‌ترین سیستم‌های رمز آلمان رمز شده بود، توانستند پیام تلگراف را رمزگشایی کرده متن آن را برای آمریکا بفرستند، بدین ترتیب آمریکا را وارد جنگ جهانی اول کرد.

اختراع و توسعه‌ی ارتباطات رادیویی حدود ۱۹۰۰ تغییری شگرف در افق رمزنگاری، به ویژه در موقعیت‌های سیاسی و نظامی فوری ایجاد کرد. حال یک ژنرال می‌تواند به طور آنی با همه‌ی سربازانش گفتگو کند، اما متاسفانه دشمن می‌تواند تمام گفتگوها را استراق سمع کند. احتیاج برای رمزهای کارا و امن بیشتر شد و منجر به اختراع رمزهای ماشینی، مثل ماشین انیگمای آلمانی شد. این ماشین دستگاهی بود که شامل تعدادی محور بود، که هر یک از آن‌ها سیم‌های زیادی داشت که از مرکز آن پخش می‌شد. پیش از آن‌که یک حرف رمز شود، محورها در جهتی تعیین شده گردش کرده، و بنابراین مسیرهای سیم‌ها و خروجی حاصل را تغییر می‌دادند. این یک رمز چند الفبایی بی‌اندازه پیچیده می‌سازد که در آن تعداد الفبای رمز بسیار بزرگ است. به علاوه، محورها می‌توانند برداشته شوند و در موقعیت‌های آغازین متفاوتی جایگزین شوند، بنابراین شکستن سیستم شامل شناخت مدارهای حول محورها و کشف موقعیت اولیه‌ی محور است.

علی‌رغم این پیچیدگی‌ها، طی جنگ جهانی دوم توانست تعداد زیادی از پیام‌هایی که با ماشین انیگما رمز شده بودند را رمزگشایی کند. آن‌ها در این تلاش از رمزنگاران هلندی کمک گرفتند، که دقیقاً پیش از آغاز جنگ، روش‌هایی که برای حمله به انیگما یافته بودند را با انگلیس و فرانسه در میان گذاشته بودند. اما هنوز هم تعیین موقعیت روزانه‌ی محورها و تحلیل جایگزینی‌شان وظیفه‌ای بینهایت سخت بود، به ویژه پس از این‌که آلمان انیگمای پیشرفته که یک محور اضافی داشت را معرفی کرد. وجود **اولترا** پروژه‌ی انگلیسی رمزگشایی انیگما تا سال ۱۹۷۴ محرمانه بود، اما امروزه گزارشات متعددی وجود دارند. هوش نظامی بدست آمده از **اولترا** در تلاش جنگ متحدین اهمیتی حیاتی داشت.

دیگر موفقیت رمزنگاری $WWII$توسط رمزنگاران آمریکایی علیه ماشین رمز ژاپنی که آن‌را پارپل نامیده بودند بدست آمد. این ماشین به جای محور از کلید استفاده می‌کرد اما باز هم اثر ایجاد یک رمز چند الفبایی پیچیده بود. گروهی از رمزنگاران، به سرپرستی ویلیام فریدمن، توانستند با تحلیل پیام‌های رمز شده‌ی بدست آمده طراحی ماشین پارپل را به طور کامل بازسازی کنند. آن‌ها سپس ماشین خود را ساختند و رهسپار رمزگشایی بسیاری از پیام‌های سیاسی مهم شدند.

در این بخش ظاهر تاریخ رمزنگاری را از عهد باستان تا اواسط قرن ۲۰ام لمس کردیم. نقاط شروع خوب برای مطالعات بیشتر شامل مقدمه‌ی سبک سیمون سینگ[۱۲۸] و کتاب سنگین و جامع، اما جذاب و خواندنی رمزشکن‌ها[۵۸] است.

§ ۷.۱ رمزهای متقارن و نامتقارن

اکنون مثال‌های مختلفی از رمز دیده‌ایم، همه‌ی آن‌ها تعدادی ویژگی مشترک دارند. باب مایل است تا یک پیام امن برای آلیس ارسال کند. او برای در هم آمیختن پیام متن ساده‌اش m و تبدیل آن به متن رمز c از کلید رمز k استفاده می‌کند. آلیس، به محض دریافت c، از کلید رمز k برای مرتب کردن c و بازسازی m استفاده می‌کند. اگر این روند به طور مناسب صورت پذیرد، آن‌گاه هم آلیس و هم باب باید کپی‌های کلید رمز k را داشته باشند، و اگر سیستم برای تامین امنیت باشد، آن‌گاه رقیب آن‌ها اِو

نباید k را بشناسد، نباید قادر به حدس زدن k باشد، و نباید قادر باشد تا بدون دانستن k پیام m را از روی c بازیابی کند.

در این بخش مفهوم یک سیستم رمز را بر اساس ریاضی محض فرمول‌بندی می‌کنیم. دلایل زیادی برای مطلوب بودن این کار وجود دارد. به ویژه، این به ما اجازه می‌دهد تا تشابه و تفاوت بین سیستم‌ها مختلف را مشخص کنیم، ضمن فراهم کردن یک چارچوب که می‌توانیم درون آن با دقت زیاد امنیت یک سیستم رمز را در قبال انواع مختلف حملات تحلیل کنیم.

§ ۱.۷.۱ رمزهای متقارن

با بازگشت به باب و آلیس، مشاهده می‌کنیم که آن‌ها باید کلید رمز k را به اشتراک بگذارند. آن‌ها می‌توانند با استفاده از آن کلید هم رمز و هم رمزگشایی کنند، بنابراین آلیس و باب اطلاع و توانایی یکسان(یا متقارن) دارند. بدین دلیل، رمزهایی از این دست به عنوان رمزهای متقارن شناخته می‌شوند. از نظر ریاضی، یک رمز متقارن از یک کلید k که از یک فضای(یعنی یک مجموعه) کلیدهای ممکن $\mathcal{K}$ انتخاب شده برای رمز کردن یک پیام متن ساده m که از فضای پیام‌های ممکن $\mathcal{M}$ انتخاب شده استفاده می‌کند، و نتیجه روند رمز کردن یک متن رمز c متعلق به فضای متون رمز ممکن $\mathcal{C}$ است.

بنابراین می‌توان رمز کردن را به عنوان یک تابع

$$e : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$$

در نظر گرفت که دامنه‌ی آن $\mathcal{K} \times \mathcal{M}$ مجموعه‌ی زوج‌های (k, m) متشکل از یک کلید k و متن ساده m و برد آن فضای متون رمزی $\mathcal{C}$ است. به طور مشابه، رمزگشایی یک تابع

$$d : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$$

است. البته، می‌خواهیم تابع رمزگشایی نتیجه تابع رمز کردن را خنثی کند. از نظر ریاضی، برای هر $k \in \mathcal{K}$ و $m \in \mathcal{M}$،

$$d(k, e(k, m)) = m.$$

گاهی مناسب است تا وابستگی به k را به عنوان زیرنویس بنویسیم. در این صورت برای هر کلید k، زوج توابع

$$e_k : \mathcal{M} \longrightarrow \mathcal{C}$$

و

$$d_k : \mathcal{C} \longrightarrow \mathcal{M}$$

را بدست می‌آوریم که در خاصیت رمزگشایی

$$d_k(e_k(m)) = m, \quad \forall m \in \mathcal{M}$$

صدق می‌کند. به عبارت دیگر، برای هر کلید k، تابع d_k تابع وارون e_k است. به ویژه، این یعنی e_k باید یک به یک باشد، زیرا اگر $e_k(m) = e_k(m')$، آن‌گاه

$$m = d_k(e_k(m)) = d_k(e_k(m')) = m'.$$

برای آلیس و باب ایمن‌تر است تا فرض کنند اِو روش رمز کردن به کار برده شده را می‌داند. به عبارت ریاضی، این یعنی اِو توابع e و d را می‌شناسد. چیزی که اِو نمی‌داند کلید خاص k است که آلیس و باب استفاده می‌کنند. برای مثال، اگر آلیس و باب از یک رمز جانشینی ساده استفاده می‌کنند، آن‌ها باید فرض کنند اِو از این حقیقت آگاه است. این قضیه‌ای پایه‌ای از رمزنگاری مدرن معروف به اصل کرشهف را نشان می‌دهد، که می‌گوید امنیت یک سیستم رمز باید تنها به اختفای کلید وابسته باشد و نه به اختفای الگوریتم رمز کردن.

اگر $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$ یک رمز موفق باشد، باید دارای خواص زیر باشد:

برای هر کلید $k \in \mathcal{K}$ و متن ساده $m \in \mathcal{M}$، محاسبه‌ی $e_k(m)$ باید آسان باشد.

برای هر کلید $k \in \mathcal{K}$ و متن رمز $c \in \mathcal{C}$، محاسبه‌ی $d_k(c)$باید آسان باشد.

برای یک یا چند متن رمز $c_1, c_2, \ldots, c_n \in \mathcal{C}$ که با استفاده از کلید $k \in \mathcal{K}$ رمز شده‌اند، محاسبه‌ی هر یک از متن‌های ساده‌ی $d_k(c_1)$، $d_k(c_2)$، ...، $d_k(c_n)$ بدون دانستن k باید مشکل باشد.

خاصیت مطلوب چهارمی نیز وجود دارد که دستیابی به آن مشکل‌تر است.

یک یا چند زوج از متون ساده و متون رمز متناظر با آن‌ها ، (m_1, c_1)، (m_2, c_2)، ... و (m_n, c_n) داده شده است، باید رمزگشایی متن رمز c که در لیست نیست بدون دانستن k مشکل باشد. از این خاصیت به عنوان امنیت علیه حمله متن ساده انتخاب شده یاد می‌شود.

مثال ۴۴.۱ رمز جانشانی ساده خاصیت ۴ را ندارد، زیرا حتی یک زوج متن ساده. متن رمز (m, c) اکثر جدول رمز را افشا می‌کند. بنابراین رموز ساده جانشانی در مقابل حملات متن ساده انتخاب شده آسیب‌پذیرند. برای مثالی بیشتر به تمرین ۴۱.۱ مراجعه کنید.

در فهرست خواص مطلوب برای یک سیستم رمز، مفهوم کلمات آسان و سخت را توضیح ندادیم. بحث تفصیلی در خصوص این سئوال عمیق را به بخش ۴.۷ موکول می‌کنیم (همچنین به بخش‌های ۲.۱ و ۲.۶ مراجعه کنید). در اینجا، به طور غیر رسمی منظور از کلمه‌ی ساده یعنی قابل محاسبه در کسری از ثانیه روی یک کامپیوتر رومیزی نوعی و سخت یعنی تمام کامپیوترهای جهان برای انجام محاسبه به (حداقل) چند سال نیاز دارند.

§ ۲.۷.۱ طرح‌های کدکردن

مناسب است تا کلیدها، متون ساده و متون رمز را به عنوان عدد در نظر گرفته و آن‌ها را به شکل دودویی بنویسیم. برای مثال، می‌توانیم رشته‌های هشت بیتی را اختیار کنیم، که اعداد بین ۰ تا ۲۵۵ را می‌دهند، و از طریق

$$a = ۰۰۰۰۰۰۰۰,\ b = ۰۰۰۰۰۰۰۱,\ c = ۰۰۰۰۰۰۱۰,\ \ldots,\ z = ۰۰۰۱۱۰۰۱$$

از آن‌ها برای نمایش حروف الفبا استفاده کنیم. برای تمایز بین حروف بزرگ و کوچک، می‌توانیم قرار دهیم $A =$ ۰۰۰۱۱۰۱۱، $B =$ ۰۰۰۱۱۱۰۰ و غیره. این روش کد کردن به ما اجازه می‌دهد تا ۲۵۶ نماد متمایز را به شکل دودویی ترجمه کنیم.

کامپیوتر شما ممکن است برای ذخیره داده‌ها از چنین روشی، معروف به کد اسکی، استفاده کند، هر چند بنا به دلایل تاریخی به کاراکترهای الفبایی کوچکترین مقادیر دودویی اختصاص داده نمی‌شود. قسمتی از کد اسکی در جدول ۷.۱ فهرست شده است.

	32	00100000
(	40	00101000
)	41	00101001
,	44	00101100
.	46	00101110

A	65	01000001
B	66	01000010
C	67	01000011
D	68	01000100
⋮	⋮	⋮
X	88	01011000
Y	89	01011001
Z	90	01011010

a	97	01100001
b	98	01100010
c	99	01100011
d	100	01100100
⋮	⋮	⋮
x	120	01111000
y	121	01111001
z	122	01111010

شکل ۱. طرح کد کردن اسکی

برای مثال، عبارت "*Bed bug.*"(شامل فاصله‌گذاری و نقطه‌گذاری) در اسکی به شکل

B	e	d		b	u	g	.
66	101	100	32	98	117	103	46
01000010	01100101	01100100	00100000	01100010	01110101	01100111	00101110

در می‌آید. پس وقتی شما عبارت "*Bed bug*" را می‌بینید، کامپیوتر شما فهرست بیت‌های

۰۱۰۰۰۰۱۰۰۱۱۰۰۱۰۱۰۱۱۰۰۱۰۰۰۰۱۰۰۰۰۰۰۱۱۰۰۰۱۰۰۱۱۱۰۱۰۱۰۱۱۰۰۱۱۱۰۰۱۰۱۱۱۰.

را می‌بینند.

تعریف ۴۵.۱ یک طرح کد کردن روشی برای تبدیل یک نوع داده به نوعی دیگر از داده است، برای مثال، تبدیل متن به اعداد. تمایز بین یک طرح کد کردن و یک طرح رمز کردن یکی از مقاصد است. فرض می‌شود یک طرح کد کردن به طور عمومی شناخته شده است و توسط هر کس برای قصد خاصی به کار می‌رود. یک طرح رمز برای پنهان کردن اطلاعات از هر کس که کلید را ندارد طراحی شده

است. بنابراین یک طرح کد کردن، مثل یک طرح رمز کردن از یک تابع کد کردن و یک تابع کدگشایی معکوس آن تشکیل شده است، هر دو تابع عمومی هستند و باید محاسبه‌ی آن سریع و آسان باشد.

با استفاده از یک طرح کد کردن، یک متن ساده یا متن رمز را می‌توان به صورت دنباله‌ای از بلوک‌های دودویی در نظر گرفت، که هر بلوک از هشت بیت تشکیل یافته، یعنی دنباله‌ای از هشت صفر و یک. یک بلوک هشت بیتی را یک بایت می‌نامند. برای درک بشری، یک بایت اغلب به عنوان یک عدد دهدهی بین ۰ و ۲۵، یا به عنوان یک عدد دورقمی شانزده شانزدهی(مبنای ۱) بین ۰۰ و FF نوشته می‌شود. کامپیوترها اغلب در یک زمان روی بیش از یک بایت عمل می‌کند. برای مثال، یک پردازنده‌ی ۶۴ بیتی در یک زمان روی هشت بایت عمل می‌کند.

§ ۳.۷.۱ رمزکردن متقارن بلوک‌های کد شده

همان‌طور که در بخش ۱.۷.۲ توضیح داده شد، در استفاده از یک طرح رمز مناسب است تا اعضای فضای متن ساده‌ی $\mathcal{M}$ را به عنوان رشته بیت‌هایی از طول ثابت B، یعنی رشته‌هایی از دقیقاً B یک و صفر در نظر گرفت. B را اندازه‌ی بلوک رمز می‌نامند. در این صورت یک پیام متن ساده معمولی از فهرستی از بلوک‌های پیام که از $\mathcal{M}$ انتخاب شده‌اند تشکیل شده، و تابع رمز کردن بلوک‌های پیام را به فهرستی از بلوک‌های متن رمز در $\mathcal{C}$ تبدیل می‌شود. که در آن هر بلوک دنباله‌ای از B بیت است. اگر متن ساده با بلوکی با کمتر از Bبیت پایان یابد، به انتهای بلوک صفر اضافه می‌کنیم. به یاد داشته باشید که این روند کد کردن، که پیام متن ساده اصلی را به دنباله‌ای از بلوک‌های بیت‌ها در $\mathcal{M}$ تبدیل می‌کند، یک دانش عمومی است.

رمزکردن و رمزگشایی در یک زمان یک بلوک انجام می‌دهند پس کافیست روند را برای یک بلوک متن، یعنی برای یک $m \in \mathcal{M}$ ساده مطالعه کنیم. این، البته، دلیل راحت بودن شکستن پیام به بلوک‌ها است. طول یک پیام می‌تواند به اندازه‌ی دلخواه بزرگ باشد، بنابراین خوب است تا بتوانیم به روند رمزکردن روی قطعه‌ای ساده از طول دلخواه تمرکز کنیم. بلوک متن ساده‌ی m رشته‌ای از B بیت است، که برای صحت آن‌را با تعداد متناظر در فرم دودویی یکی می‌گیریم. به عبارت دیگر، $\mathcal{M}$ را به وسیله‌ی

$$\overbrace{m_{B-1}m_{B-2}\cdots m_2m_1m_0}^{\text{list of } B \text{ bits of } m} \longleftrightarrow \overbrace{m_{B-1}\cdot 2^{B-1}+\cdots+m_2\cdot 2^2+m_1\cdot 2+m_0}^{\text{integer between 0 and } 2^B-1}$$

با مجموعه‌ی اعداد صحیح m که $۰ \leq m < ۲^B$ یکی می‌گیریم، در اینجا $m_۰$، $m_۱$، ...، $m_{B-۱}$ صفر یا یک هستند.

به طور مشابه، فضای کلید $\mathcal{K}$ و فضای متن رمز $\mathcal{C}$ را با اعداد صحیح متناظر با رشته‌های بیتی از طول بلوک معین یکی می‌گیریم. برای تسهیل در نوشتن، طول بلوک برای کلیدها، متون ساده و متون رمز را با B_k، B_m و B_c نمایش می‌دهیم. آن‌ها لزوماً یکی نیستند. بنابراین $\mathcal{K}$، $\mathcal{M}$ و $\mathcal{C}$ را با مجموعه‌های اعداد صحیح مثبت

$$\mathcal{K} = \{k \in \mathbb{Z} : ۰ \leq k < ۲^{B_k}\}$$

$$\mathcal{M} = \{m \in \mathbb{Z} : ۰ \leq m < ۲^{B_m}\}$$

$$\mathcal{C} = \{c \in \mathbb{Z} : ۰ \leq c < ۲^{B_c}\}$$

یکی می‌گیریم. فوراً سئوالی مهم به وجود می‌آید: آلیس و باب مجموعه‌ی $\mathcal{K}$ را باید تا چه اندازه بزرگ انتخاب کنند، به عبارت دیگر، اندازه بلوک کلید B_k را باید چقدر بزرگ انتخاب کنند؟ اگر B_k خیلی کوچک باشد، آن‌گاه اِو می‌تواند هر عدد بین ۰ تا $۲^{B_k} - ۱$ را بررسی کند تا کلید آلیس و باب را بیابد. به طور دقیق‌تر، از آن‌جا که فرض می‌شود اِو الگوریتم رمزگشایی d (اصل کیرشهف) را می‌داند، هر $k \in \mathcal{K}$ را برداشته و از آن برای محاسبه‌ی $d_k(c)$ استفاده می‌کند. با این فرض که اِو قادر است متون ساده درست و نادرست را تشخیص دهد، در نهایت قادر به بازیابی متن خواهد بود.

این حمله به عنوان یک حمله جستجوی جامع(گاهی اوقات نیز از آن با نام حمله‌ی بروت- فورس نیز یاد می‌شود)، زیرا اِو در فضای کلید جستجوی جامع انجام می‌دهد. با تکنولوژی امروزی، جستجوی جامع وقتی فضا حداقل $۲^{۸۰}$ عضو دارد غیرکارآمد فرض می‌شود. بنابراین باب و آلیس قطعاً $B_k \geq ۸۰$ را انتخاب می‌کند.

برای بسیاری از سیستم‌های رمزنگاری، به ویژه سیستم‌های رمزنگاری کلید عمومی که هسته‌ی

این کتاب را تشکیل می‌دهند، تظریف‌هایی برای حملات جستجوی جامع وجود دارند که به طور موثر اندازه‌ی فضا را با ریشه‌ی دوم آن جایگزین می‌کنند. این روش‌ها بر پایه‌ی این اصل است که یافتن اشیاء یکسان(تصادم) در یک مجموعه آسان‌تر از یافتن یک شی خاص است. برخی از این حملات تصادم یا ملاقات-در-میان راه را در بخش‌های ۲۰، ۴۰، ۴۰، ۶.۲ و ۶.۱۰ توضیح می‌دهیم. اگر حملات ملاقات-در-میان راه وجود داشته باشند، آن‌گاه آلیس و باب باید $B_k \geq ۱۶۰$ را انتخاب کند.

§ ۴.۷.۱ مثال‌هایی از رمزهای متقارن

پیش از اینکه بیش از این در باتلاق نظری و نمادگذاری فرو رویم، توقفی داریم تا توصیف ریاضی برخی رمزهای متقارن مقدماتی را ارائه کنیم.

فرض کنید p یک عدد اول بزرگ باشد، برای مثال $۲^{۱۵۹} < p < ۲^{۱۶۰}$. آلیس و باب فضای کلید $\mathcal{K}$، فضای متون $\mathcal{M}$، و فضای متون رمز $\mathcal{C}$ را مجموعه‌ی

$$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{۱, ۲, ۳, \ldots, p-۱\}$$

انتخاب می‌کنند. در اصطلاح خیالی، $\mathcal{K} = \mathcal{M} = \mathcal{C} = \mathbb{F}_p^*$ همگی برابر گروه یکال‌ها در میدان متناهی $\mathbb{F}_p$ اختیار می‌شوند.

آلیس و باب به طور تصادفی یک کلید $k \in \mathcal{K}$، یعنی یک عدد صحیح k را انتخاب می‌کنند که $۱ \leq k < p$، و تصمیم می‌گیرند از تابع رمز کردن e_k که به صورت زیر تعریف شده استفاده کنند.

$$e_k(m) \equiv k \cdot m \pmod{p}. \tag{۹.۱}$$

در اینجا $e_k(m)$ برابر است با مجموعه‌ی اعداد صحیح مثبت بین ۱ و p که به پیمانه‌یp با $k.m$ برابرند. تابع رمزگشایی متناظر d_k برابر است با

$$d_k(c) \equiv k' \cdot c \pmod{p},$$

که k' وارون k به پیمانه‌ی p است. یادآوری این نکته مهم است که هر چند p بسیار بزرگ است،

الگوریتم توسعه یافته‌ی اقلیدس(تذکر ۲۰.۱) به ما اجازه می‌دهد در کمتر از $2log_2p + 2$ گام k' را محاسبه کنیم. بنابراین یافتن k' از k در شمار ساده‌های جهان رمزنگاری قرار می‌گیرد.

واضح است که حدس زدن k برای اِو مشکل است، زیرا تقریباً 2^{160} حالت برای انتخاب وجود دارد. آیا اگر او متن رمزی c را بشناسد بازیابی k از آن نیز کار دشواری خواهد بود؟ اگر پاسخ مثبت است، این کار همچنان مشکل خواهد بود. توجه کنید که تابع رمز

$$e_k : \mathcal{M} \longrightarrow \mathcal{C}$$

برای هر کلید انتخابی k پوشا(به رو) است. این یعنی برای هر $c \in \mathcal{C}$ و $k \in \mathcal{K}$، $m \in \mathcal{M}$ موجود است که $e_k(m) = c$. به علاوه، متن رمزی ممکن است هر متن ساده‌ای را نمایش دهد، به این شرط که با یک کلید مناسب رمز شده باشد. به زبان ریاضی، می‌توان گفت برای هر متن رمزی $c \in \mathcal{C}$ و متن ساده $m \in \mathcal{M}$، کلید k موجود است به نحوی که $e_k(m) = c$. به ویژه این مطلب برای کلید

$$k \equiv m^{-1} \cdot c \pmod p \qquad (۱۰.۱)$$

درست است. این نشان می‌دهد رمز آلیس و باب خواص ۱،۲ و ۳ که در صفحه‌ی ؟؟ آمد را دارد، زیرا هر کس که k را بداند می‌تواند به راحتی رمز کرده و رمزگشایی کند، اما اگر مقدار k را نداند رمزگشایی سخت است. هر چند، این رمز خاصیت ۴ را ندارد، زیرا حتی یک زوج متن ساده.متن رمز (m, c) به اِو اجازه می‌دهد تا با استفاده از فرمول (۱۰.۱) k، را بازیابی کند.

همچنین جالب است که مشاهده کنیم اگر آلیس و باب تابع رمز کردنشان را به سادگی ضرب اعداد صحیح $e_k(m) = k \cdot m$ بدون تحویل به پیمانه‌ی p تعریف کنیم، آن‌گاه رمز آنها هنوز خواص ۱ و ۲ را دارند، اما خاصیت ۳ برقرار نیست. اگر اِو سعی کند تا یک متن رمز ساده $c = k \cdot m$ را رمزگشایی کند، هنوز با وظیفه‌ی سخت تجزیه‌ی یک عدد بزرگ مواجه است. هرچند، اگر او موفق شود چند متن رمز c_1، c_2، ...، c_n را کسب کند، آن‌گاه شانسی خوب وجود دارد که

$$gcd(c_1, c_2, \ldots, c_n) = gcd(k \cdot m_1, k \cdot m_2, \ldots, k \cdot m_n) \qquad (۱۱.۱)$$

$$= k \cdot gcd(m_1, m_2, \ldots, m_n) \qquad (۱۲.۱)$$

برابر با خود k یا مضرب کوچکی از k باشد. توجه کنید که محاسبه‌ی بزرگترین مقسوم علیه مشترک کاری آسان است.

این مشاهده اولین نشانه از این است که چگونه تحویل به پیمانه‌ی p تاثیر مخلوط شگفت آوری دارد که خواصی مثل تقسیم پذیری را خراب می‌کند. هرچند، تحویل به خودی خود پاسخ نهایی نیست. آسیب‌پذیری رمز (؟؟) نسبت به حمله متن رمز انتخابی را در نظر بگیرید. همان‌طور که در بالا اشاره شد، اگر اِو بتواند به هر دو متن رمز c و متن ساده متناظر m دست یابد، آن‌گاه اِو می‌تواند با محاسبه‌ی

$$k \equiv m^{-1} \cdot c \pmod p$$

کلید را بازیابی کند. بنابراین حتی یک زوج متن رمز.متن ساده برای افشای کلید کافیست، بنابراین تابع رمز کردن e_k که با (؟؟) داده شده خاصیت ۴ صفحه؟؟ را ندارد.

انواع مختلفی از این رمز ضرب-به پیمانه-p وجود دارد. برای مثال، از آن‌جا که جمع بسیار کاراتر از ضرب کردن است، یک رمز جمع-به-پیمانه‌ی p وجود دارد که با

$$e_k(m) \equiv m + k \pmod p$$

و

$$d_k(c) \equiv c - k \pmod p$$

داده می‌شود، که چیزی جز رمز انتقالی یا رمز قیصری که در بخش ۱.۱ مطالعه کردیم نیست. نوع دیگر که رمز آفین خوانده می‌شود، تلفیق رمز انتقالی و رمز حاصل‌ضرب است. کلید یک رمز آفین از دو عدد صحیح $k = (k_1, k_2)$ تشکیل شده و رمز کردن و رمزگشایی به شکل

$$\begin{aligned} e_k(m) &= k_1 \cdot m + k_2 \pmod p, \\ d_k(c) &= k'_1 \cdot (c - k_2) \pmod p, \end{aligned} \tag{۱۳.۱}$$

تعریف می‌شوند، که در آن k'_1 معکوس k_1 به پیمانه‌ی p است.

رمز آفین تعمیمی معروف به رمز هیل دارد، که در آن متن ساده m، متن رمز c، و بخش دوم کلید $k_۲$ با بردارهای ستونی متشکل از n عدد به پیمانه‌ی p جایگزین شده‌اند. اولین قسمت کلید $k_۱$ یک ماتریس n در n با درایه‌هایی به پیمانه‌ی p انتخاب می‌شود. رمز کردن و رمز گشایی باز با استفاده از (۱۳.۱) تعریف می‌شوند، با این تفاوت که ضرب $k_۱ \cdot m$ حاصل‌ضرب یک ماتریس و یک بردار است، و $k'_۱$ ماتریس وارون $k_۱$ به پیمانه‌ی p است. هر دو رمز آفین و هیل در مقابل حملات متون انتخابی آسیب‌پذیر هستند(به تمرین ۴۱.۱ و ۴۲.۱ مراجعه کنید).

مثال ۴۶.۱ همان‌طور که پیشتر اشاره شد، در حالت کلی جمع سریع‌تر از ضرب است، اما عملگر کامپیوتری دیگری نیز وجود دارد که حتی از ضرب سریع‌تر است. این عملگر را یای انحصاری نامیده و آن‌را با XOR یا $\oplus$ نمایش می‌دهیم. در کوچکترین سطح، XOR دو بیت تکی $\beta \in \{۰, ۱\}$ و $\beta' \in \{۰, ۱\}$ را گرفته و

$$\beta \oplus \beta' = \begin{cases} ۰ \ \ (mod\ p) & \text{اگر } \beta \text{ و } \beta' \text{ یکی باشند} \\ ۱ \ \ (mod\ p) & \text{اگر } \beta \text{ و } \beta' \text{ متفاوت باشند} \end{cases} \tag{۱۴.۱}$$

اگر یک بیت را به عنوان یک عدد ۰ یا ۱ در نظر بگیرد، آن‌گاه XOR همان جمع به پیمانه‌ی ۲ است. به طور کلی‌تر XOR دو رشته‌ی بیتی نتیجه اعمال XOR روی هر زوج متناظر از بیت‌هاست. برای مثال،

$$۱۰۱۱۰ \oplus ۱۱۰۱۰ = [۱ \oplus ۱][۰ \oplus ۱][۱ \oplus ۰][۱ \oplus ۱][۰ \oplus ۰] = ۰۱۱۰۰.$$

با استفاده از این عملگر جدید، آلیس و باب رمز پایه‌ای دیگری در دسترس دارند که به صورت

$$e_k(m) = k \oplus m, \qquad d_k(c) = k \oplus c$$

تعریف می‌شوند. در اینجا $\mathcal{K}$، $\mathcal{M}$ و $\mathcal{C}$ مجموعه‌های همه‌ی رشته‌های دودویی از طول B، یا به طور معادل، مجموعه‌ی همه‌ی اعداد بین ۰ و $۲^B - ۱$ هستند.

کارایی بالا و کاملاً تقارنی بودن، بدین معنی که e_k و d_k یک تابع هستند، از مزایای این رمز است. اگر k به طور تصادفی انتخاب شوند و تنها یک‌بار مصرف شود، آن‌گاه این رمز به عنوان پد یکبار مصرف ورنام شناخته می‌شود. در بخش ۴.۵۶ نشان می‌دهیم که پد یکبار مصرف به طور اثبات پذیر امن هستند. متاسفانه، به کلیدی با طول متن ساده احتیاج است، که آن‌را برای بسیاری از کاربردهای عملی طاقت فرسا می‌کند. و اگر k برای رمز کردن بیش از یک متن ساده استفاده شود، آن‌گاه اِو ممکن است برای کسب اطلاعات در خصوص m یا m' از این حقیقت استفاده کند که

$$c \oplus c' = (k \oplus m) \oplus (k \oplus m') = m \oplus m'.$$

روشن نیست اِو برای یافتن k، m و m' چگونه پیش می‌رود، اما این حقیقت که کلید k می‌تواند به سادگی حذف شود، مقدار $m \oplus m'$ را افشا می‌کند که کمتر تصادفی است، باید رمزنگار را عصبی کند. به علاوه، این روش در برخی موارد نسبت به حمله متن انتخابی آسیب‌پذیر است؛ به تمرین ۴۶.۱ مراجعه کنید.

§ ۵.۷.۱ دنباله‌های بیتی تصادفی و رمزهای متقارن

پیشتر، به سئوال اساسی مربوط به ایجاد رمزهای متقارن کارا و امن رسیده‌ایم. آیا ممکن است از یک کلید نسبتاً کوچک k(برای مثال متشکل از ۱۶۰ بیت تصادفی) یک پیام به دلخواه بزرگ را به طور امن و کارا ارسال کرد؟ در اینجا یک ساختار ممکن وجود دارد. فرض کنید می‌توانیم یک تابع

$$R : \mathcal{K} \times \mathbb{Z} \longrightarrow \{۰, ۱\}$$

با خواص زیر را بسازیم:

برای هر $k \in \mathcal{K}$ و $j \in \mathbb{Z}$، محاسبه‌ی $R(k, j)$ آسان است.

برای دنباله‌ی به اندازه‌ی دلخواه بزرگ از اعداد صحیح $j_۱$، $j_۲$، ... و j_n و تمام مقادیر $R(k, j_۱)$، $R(k, j_۲)$، ... و $R(k, j_n)$داده شده، تعیین kمشکل باشد.

برای هر لیست داده شده از اعداد صحیح j_1، j_2، ... و j_n و مقادیر $R(k, j_1)$، $R(k, j_2)$، ... و $R(k, j_n)$، شانس درست حدس زدن $R(k, j)$ برای jی که در لیست قرار ندارد کمتر از ٪۵۰ باشد.

اگر بتوانیم یک تابع R با این سه خاصیت بیابیم، آن‌گاه می‌توانیم از آن برای تبدیل یک کلید k به دنباله‌ای از بیت‌های

$$R(k, 1), R(k, 2), R(k, 3), R(k, 4), \ldots, \tag{۱۵.۱}$$

استفاده کرده و سپس می‌توانیم همان طور که در مثال ۴۶.۱ توضیح دادیم، از این دنباله بیت‌ها به عنوان کلید برای یک پد یکبار مصرف استفاده کنیم.

مسئله‌ی اساسی در این رویکرد این است که دنباله‌ی بیت‌های (۱۵.۱) حقیقتاً تصادفی نیستند، زیرا توسط تابع R تولید شده‌اند. در عوض، می‌گوییم دنباله‌ی بیت‌های (۱۵.۱) دنباله‌ای شبه تصادفی است و R را یک مولد اعداد شبه تصادفی می‌خوانیم.

آیا مولدهای اعداد شبه تصادفی وجود دارند؟اگر چنین باشد، مثال‌هایی از توابع یک طرفه فراهم می‌کنند که دیفیه هلمن در مقاله‌ی خود [۳۶] تعریف کرده‌اند، اما علیرغم بیش از ربع قرن تلاش، هیچ کس نتوانسته وجود حتی یک چنین تابعی را اثبات کند. در بخش‌های ۲.۱ و ۸.۲ به این موضوع جذاب باز خواهیم گشت. برای حال، به چند تذکر اکتفا می‌کنیم.

هر چند هیچ کس نتوانسته به طور قطعی وجود مولدهای اعداد تصادفی را اثبات کند، نامزدهای زیادی پیشنهاد شده‌اند، و برخی از این پیشنهادها در برابر آزمون زمان مقاومت کرده‌آند. دو رویکرد پایه‌ای برای ساختن نامزدهای R وجود دارد، و این دو روش تناقض اساسی بین امنیت و کارایی در رمزنگاری را به خوبی توصیف می‌کنند.

رویکرد اول به کار بردن مکرر خانواده‌ای تک موردی عملکردهای مخلوطیست که محاسبه‌ی آن‌ها کاراست و حل آن‌ها بسیار سخت است. این روش پایه‌ی بسیاری از رمزهای متقارن عملی، شامل استاندارد رمز کردن داده‌ها DES و استاندارد رمز کردن پیشرفته AES است، که دو سیستم متداول

امروزی هستند. برای مطالعه خلاصه‌ای از این رمزهای متقارن امروزی به بخش ۸.۱۰ مراجعه کنید.

رویکرد دوم ساختن R با استفاده از تابعی است که وارون کارای آن یک مسئله ریاضی مشهور سخت است. این رویکرد یک زیربنای نظری بسیار رضایت بخش برای یک رمز متقارن است، اما متاسفانه، کارایی ساختارهایی از این دست بسیار کمتر از ساختارهای $ad\ hoc$ است، و بنابراین برای کاربردهای واقعی جذابیت کمتری دارند.

§ ۶.۷.۱ رمزهای نامتقارن اولین ظهور را می‌سازند

اگر آلیس و باب مایل به تبادل کلید با استفاده از یک رمز متقارن باشند، آن‌ها در ابتدا باید متقابلاً روی یک کلید مخفی k به توافق برسند. اگر آن‌ها فرصت ملاقات در خفا را داشته باشند یا بتوانند یکبار از طریق یک کانال ارتباطی امن مکالمه کنند خوب است. اما اگر این فرصت را نداشته باشند و هر ارتباط بین آن‌ها به گوش اِو برسد چه؟ آیا ممکن است آلیس و باب تحت این شرایط یک کلید مخفی را مبادله کنند؟

اولین عکس العمل افراد این است که ممکن نیست، زیرا اِو همه‌ی قطعات اطلاعاتی که آلیس و باب رد و بدل می‌کنند را می‌بیند. این پرتو تابان دیفیه هلمن بود که تحت مفروضات خاصی، این کار ممکن است. جستجو برای پاسخ‌های کارا(و قابل اثبات) برای این مسئله، که رمزنگاری کلید عمومی(یا نامتقارن) خوانده می‌شود، یکی از جذاب‌ترین قسمت‌های رمزنگاری ریاضی را تشکیل می‌دهد و تمرکز اصلی این کتاب است.

برای تجسم رمزنگاری کلید عمومی با توصیف یک راه غیر ریاضی شروع می‌کنیم. آلیس یک گاو صندوق با یک سوراخ باریک تهیه کرده و آن‌را در معرض عموم قرار می‌دهد. هر کس در جهان اجازه دارد صندوق را بررسی کرده و امنیت آن‌را ببیند. باب پیام خود برای آلیس را روی قطعه‌ای کوچک نوشته و از سوراخ به داخل صندوق می‌اندازد. حال تنها کسی که کلید را داشته باشد، که احتمالاً تنها آلیس است، می‌تواند پیام باب را بخواند. در این سناریو، کلید عمومی آلیس گاو صندوق است، الگوریتم رمز کردن روند قرار دادن پیام در صندوق است و الگوریتم رمزگشایی روند باز کردن صندوق

با کلید است. توجه کنید این ترتیب بعید است؛ این در دنیای واقعی استفاده می‌شود. برای مثال، the slot deposit night در بانک بدین شکل است، هر چند در عمل شکاف باید به خوبی حفاظت شده باشد تا از ورود یک تنبر باریک و باز کردن سپرده دیگران جلوگیری شود!

جنبه‌ی مفید سیستم رمزنگاری صندوق-با-یک-شکاف، که با سیستم رمزنگاری کلید عمومی مشترک است، این است که آلیس تنها به یک شکاف در مکان عمومی احتیاج دارد، و سپس هر کس در جهان می‌تواند مکرراً از آن برای ارسال پیام به آلیس استفاده کند. آلیس نیازی ندارد برای هر یک از طرفینش یک شکاف مجزا تهیه کند. و نیز لازم نیست صندوق را باز کرده و شخص دیگری مثل کارل یا دیو از آن برای ارسال پیام به آلیس استفاده کنند، پیام باب را بخواند.

حال آماده‌ایم تا فرمول‌بندی ریاضی یک رمز نامتقارن را ارائه کنیم. طبق معمول، فضای کلیدهای $\mathcal{K}$، متون ساده $\mathcal{M}$ و متون رمز $\mathcal{C}$ وجود دارند. هر چند عضو k از فضای کلید در حقیقت یک زوج کلیدهای

$$k = (k_{priv}, k_{pub}),$$

است که به ترتیب کلید خصوصی و کلید عمومی خوانده می‌شوند. برای هر کلید عمومی k_{pub} تابع رمز کردن متناظر

$$e_{k_{pub}} : \mathcal{M} \longrightarrow \mathcal{C}$$

وجود دارد، و برای هر کلید خصوصی k_{priv} تابع رمزگشایی متناظر

$$d_{k_{priv}} : \mathcal{C} \longrightarrow \mathcal{M}$$

وجود دارد. این توابع دارای خاصیت هستند که اگر زوج (k_{priv}, p_{pub}) در فضای کلید $\mathcal{K}$ باشند، آن‌گاه برای هر $m \in \mathcal{M}$،

$$d_{k_{priv}}(e_{k_{pub}}(m)) = m.$$

برای امن بودن یک رمز متقارن، باید محاسبه‌ی تابع رمزگشایی $d_{k_{priv}}(c)$ برای اِو مشکل باشد، حتی در حالتی که کلید عمومی $e_{k_{pub}}$ را دارد. توجه داشته باشید که تحت این فرض، آلیس می‌تواند با استفاده از یک کانال ارتباطی ناامن k_{pub} را برای باب ارسال کند، و باب می‌تواند متن رمز $e_{k_{pub}}(m)$ را باز بفرستد، بدون نگرانی از اینکه اِو قادر به رمزگشایی پیام خواهد بود. برای رمزگشایی آسان، شناخت کلید خصوصی k_{priv} لازم است، و احتمالاً آلیس تنها شخص با این اطلاعات است. کلید خصوصی را گاهی اوقات اطلاعات دریچه آلیس می‌نامند، زیرا یک دریچه(یعنی یک میان‌بر) برای محاسبه‌ی تابع معکوس $e_{k_{pub}}$ فراهم می‌کند. این حقیقت که کلیدهای عمومی و خصوصی k_{pub} و k_{priv} متفاوت هستند رمز را نامقارن می‌کند، و این دلیل نام‌گذاری آن است.

این $quiteintriguing$ است که دیفایه و هلمن بدون یافتن یک کاندید برای یک زوج حقیقی از توابع این مفهوم را ساختند، هرچند آن‌ها روشی مشابه پیشنهاد کردند که آلیس و باب می توانند از آن برای تبادل امن قطعه‌ای تصادفی از داده‌هایی که مقدارش در ابتدا برای دیگری شناخته شده نیست استفاده کنند. روش تبادل کلید دیفیه و هلمن را در بخش ۲.۳ توضیح داده و سپس در ادامه به توصیف تعدادی رمز نامتقارن، شامل الجمال(بخش ۲)، RSA(بخش ۳)، ECC(بخش ۵)، و $NTRU$ (بخش ۶.۱) می‌پردازیم، که امنیت آن‌ها بر مبنای سختی احتمالی چند مسئله‌ی ریاضی متفاوت است.

§ تمرین

۱.۱ یک چرخنده رمزی مثل آنکه در شکل ۱.۱ توضیح داده شده بسازید، با این تفاوت که چرخ داخلی گردنده باشد، و از آن برای انجام کارهای زیر استفاده کنید. *(برای راحتی شما، یک چرخنده رمزی در $www.math.brown.edu/\ jhs/MathCrypto/CipherWheel.pdf$ قرار دارد که می‌توانید آن‌را چاپ کرده و ببرید.)*

با استفاده از دوران به اندازه در جهت گردش ساعت متن ساده زیر را رمز کنید.

A page of history is worth a volume of logic.

متن زیر با دوران به اندازه‌ی در جهت گردش ساعت رمز شده است، آن‌را رمزگشایی کنید.

AOLYLHYLUVZLJYLAZILAALYAOHUAOLZLJYLALZAOHALCLYFIVKFNBLZZLZ

متن زیر را رمزگشایی کنید، که در آن حرف اول به اندازه یکی در جهت گردش ساعت، دومی به اندازه دوتا در جهت گردش ساعت، و غیره رمز شده است.

XJHRFTNZHMZGAHIUETXZJNBWNUTRHEPOMDNBJMAUGORFAOIZOCC

۲.۱ با آزمودن انتقال‌های ممکن متعددهر یک از متن‌های زیر که با استفاده از رمزهای قیصر رمز شده‌اند را رمزگشایی کنید تا به یک متن معنادار برسید.

LWKLQNWKDWLVKDOOQHYHUVHHDELOOERDUGORYHOBDVDWUHH

UXENRBWXCUXENFQRLQJUCNABFQNWRCJUCNAJCRXWORWMB

BGUTBMBGZTFHNLXMKTIPBMAVAXXLXTEPTRLEXTOXKHHFYHKMAXFHNLX

۳.۱ برای این تمرین، از جدول جایگزینی داده شده در جدول ؟؟ استفاده کنید.

پیام متن ساده‌ی

The gold is hidden in the garden

را رمز کنید.

یک جدول رمزگشایی بسازید، یعنی، جدولی بسازید که الفبای رمزی به ترتیب از A تا Z باشند و الفبای متن ساده درهم باشند.

از جدول رمزگشایی‌تان که در قسمت (ب) ساختید برای رمزگشایی پیام زیر استفاده کنید.

IBXLX JVXIZ SLLDE VAQLL DEVAU QLB

۴.۱ هر یک از متون زیر زیر با استفاده از یک رمز جانشانی ساده رمز شده‌اند. برای راحتی شما، یک جدول فراوانی و فهرستی از دو حرفی‌های متداولی که در متن رمز آمده‌اند به شما داده‌ایم.(اگر نمی‌خواهید متون رمزی را با دست کپی کنید، می‌توانید آن را از وب‌سایت فهرست شده در مقدمه دانلود یا چاپ کنید.)

"A Piratical Treasure"

JNRZR BNIGI BJRGZ IZLQR OTDNJ GRIHT USDKR ZZWLG OIBTM
NRGJN IJTZJ LZISJ NRSBL QVRSI ORIQT QDEKJ JNRQW GLOFN
IJTZX QLFQL WBIMJ ITQXT HHTBL KUHQL JZKMM LZRNT OBIMI
EURLW BLQZJ GKBJT QDIQS LWJNR OLGRI EZJGK ZRBGS MJLDG
IMNZT OIHRK MOSOT QHIJL QBRJN IJJNT ZFIZL WIZTO MURZM
RBTRZ ZKBNN LFRVR GIZFL KUHIM MRIGJ LJNRB GKHRT QJRUU
RBJLW JNRZI TULGI EZLUK JRUST QZLUK EURFT JNLKJ JNRXR
S

این متن رمز شامل ۳۱۶ حرف است. در اینجا یک جدول فراوانی هست:

	B	R	G	N	A	I	U	K	O	J	L	X	M	F	S	E	Z	C	T	W	P	V	Q
Freq	32	28	22	20	16	16	14	13	12	11	10	10	8	8	7	7	6	5	3	2	1	1	1

فراوان‌ترین دو حرفی‌ها عبارتند از: JN (۱۱ بار)، NR (بار)، TQ (بار)، و LW، RB، RZ و JL (هر کدام ۵ بار).

"A Botanical Code"

KZRNK GJKIP ZBOOB XLCRG BXFAU GJBNG RIXRU XAFGJ BXRME MNKNG BURIX KJRXR SBUER ISATB UIBNN RTBUM NBIGK EBIGR OCUBR GLUBN JBGRL SJGLN GJBOR ISLRS BAFFO AZBUN RFAUS AGGBI NGLXM IAZRX RMNVL GEANG CJRUE KISRM BOOAZ GLOKW FAUKI NGRIC BEBRI NJAWB OBNNO ATBZJ KOBRC JKIRR NGBUE BRINK XKBAF QBROA LNMRG MALUF BBG

متن رمز شامل ۲۵۳ حرف است، در اینجا جدول فراوانی است:

	B	R	G	N	A	I	U	K	O	J	L	X	M	F	S	E	Z	C	T	W	P	V	Q
Freq	32	28	22	20	16	16	14	13	12	11	10	10	8	8	7	7	6	5	3	2	1	1	1

فراوان‌ترین دو حرفی‌ها عبارتند از: NG و RI (هر کدام ۷ بار)، BU (۶ بار) و ER (۵ بار).

برای مشکل‌تر کردن این تمرین، تمام کلمات the را از متن ساده حذف می‌کنیم.

"A Brilliant Detective"

GSZES GNUBE SZGUG SNKGX CSUUE QNZOQ EOVJN VXKNG XGAHS AWSZZ BOVUE SIXCQ NQESX NGEUG AHZQA QHNSP CIPQA OIDLV JXGAK CGJCG SASUB FVQAV CIAWN VWOVP SNSXV JGPCV NODIX GJQAE VOOXC SXXCG OGOVA XGNVU BAVKX QZVQD LVJXQ EXCQO VKCQG AMVAX VWXCG OOBOX VZCSO SPPSN VAXUB DVVAX QJQAJ VSUXC SXXCV OVJCS NSJXV NO-JQA MVBSZ VOOSH VSAWX QHGMV GWVSX CSXXC VBSNV

ZVNVN SAWQZ ORVXJ CVOQE JCGUW NVA

متن رمزی شامل ۳۱۳ حرف است. در اینجا یک جدول فراوانی آمده است:

	V	S	X	G	A	O	Q	C	N	J	U	Z	E	W	B	P	I	H	K	D	M	L	R	F
Freq	39	29	29	22	21	21	20	20	19	13	11	11	10	8	8	6	5	5	5	4	3	2	1	1

فراوان‌ترین دو حرفی‌ها عبارتند از: XC (۱۰ بار)، NV (۷ بار)، و CS، QV، QA و SX (هر یک ۶ بار).

۵.۱ فرض کنید الفبایی ۲۶ حرفی دارید.

چند رمز جانشینی ساده ممکن وجود دارند؟

یک حرف الفبا را ثابت می‌گوییم اگر پس از رمز کردن ثابت می‌ماند. چند رمز جانشینی ساده وجود دارند که:

- هیچ حرفی را ثابت نگه ندارد؟
- حداقل یک حرف ثابت بماند؟
- دقیقاً یک حرف ثابت بماند؟
- حداقل دو حرف ثابت بماند؟

(قسمت (ب) مشکل‌تر است! شما ابتدا باید با الفبایی چهار یا پنج حرفی آغاز کنید تا ایده‌ی کار را بدست آورید.)

بخش ۱.۲ بخش‌پذیری و بزرگترین مقسوم علیه مشترک

۶.۱ فرض کنید $a, b, c \in \mathbb{Z}$. با استفاده از تعریف بخش‌پذیری خاصیت‌های بخش‌پذیری زیر را اثبات کنید. (این گزاره‌ی ۱.۴ است.)

اگر $a|b$ و $b|c$ آن‌گاه $a|c$.

اگر $a|b$ و $b|a$ آن‌گاه $a = \pm b$.

اگر $a|b$ و $a|c$ آن‌گاه $a|b+c$ و $a|b-c$.

۷.۱ از یک ماشین حساب و روش توضیح داده شده در تذکر ۱۲.۱ برای محاسبه‌ی خارج قسمت و باقیمانده‌های زیر استفاده کنید.

۳۴۷۸۷ تقسیم بر ۳۵.

۲۳۸۷۹۲ تقسیم بر ۷۸۴.

۹۸۲۹۳۸۷۴۹۳ تقسیم بر ۸۷۳۴۸.

۱۴۹۸۳۸۷۴۸۷ تقسیم بر ۷۶۳۴.

۸.۱ با استفاده از یک ماشین حساب و روش توضیح داده شده در تذکر ۱۲.۱، بدون دردسر محاسبه‌ی خارج قسمت وابسته باقیمانده‌های زیر را محاسبه کنید.

باقیمانده تقسیم ۷۸۷۴۵ بر ۱۲۷.

باقیمانده تقسیم ۲۸۳۷۶۴۷ بر ۴۳۸۷.

باقیمانده تقسیم ۸۷۳۹۲۸۷۴۶۳ بر ۱۸۷۵۴.

باقیمانده تقسیم ۴۵۳۶۷۸۲۷۹۳ بر ۹۷۸۴۵۳۷.

۹.۱ از الگوریتم اقلیدس برای محاسبه‌ی بزرگترین مقسوم علیه‌های مشترک زیر استفاده کنید.

$gcd(۲۹۱, ۲۵۲)$.

$gcd(۱۶۲۶۱, ۸۵۶۵۲)$.

$gcd(۱۳۹۰۲۴۷۸۹, ۹۳۲۷۸۸۹۰)$.

$gcd(۱۶۵۳۴۵۲۸۰۴۴, ۸۳۳۲۷۴۵۹۲۷)$.

۱۰.۱ برای هر یک از مقادیر $gcd(a, b)$ از تمرین ۹.۱ با استفاده از الگوریتم اقلیدس مقادیر صحیح u و v را به نحوی بیابید که $au + bv = gcd(a, b)$.

۱۱.۱ فرض کنید a و b اعداد صحیح باشند.

فرض کنید اعداد صحیح u و v موجود باشند که $au + bv = ۱$. ثابت کنید $gcd(a, b) = ۱$.

فرض کنید اعداد صحیح u و v موجود باشند که $au + bv = ۶$. آیا لزوماً $gcd(a, b) = ۶$؟ اگر نه، یک مثال نقض ارائه کرده، و تمام مقادیر ممکن برای $gcd(a, b)$ را توصیف کنید.

فرض کنید $(u_۱, v_۱)$ و $(u_۲, v_۲)$ جواب‌های صحیح معادله‌ی $au + bv = ۱$ باشند. ثابت کنید a، $v_۲ - v_۱$ را عاد می‌کند و b، $u_۲ - u_۱$ را عاد می‌کند.

به طور کلی‌تر، فرض کنید $g = gcd(a, b)$ و $(u_۰, v_۰)$ جواب صحیح $au+bv = g$ باشد. ثابت کنید هر جواب دیگر به شکل $u = u_۰ + kb/g$ و $v = v_۰ - ka/g$ است که در آن k یک عدد صحیح است.(این قسمت دوم قضیه‌ی ۱۴.۱ است.)

۱۲.۱ روش حل $au + bv = gcd(a, b)$ که در بخش ۱.۲ توضیح داده شد، تا حدودی ناکارآمد است. این تمرین روشی برای محاسبه‌ی u و v توضیح می‌دهد که برای پیاده‌سازی کامپیوتری بسیار مناسب است. به ویژه فضای کمی لازم دارد.

نشان دهید الگوریتم زیر بزرگترین مقسوم علیه مشترک g از اعداد صحیح مثبت a و b به همراه یک جواب صحیح (u, v) برای معادله‌ی $au + bv = g$ را محاسبه می‌کند.

(آ) قرار بده $u = ۱$، $g = a$، $x = ۰$ و $y = b$.

(ب) اگر $y = ۰$، قرار بده $v = (g - au)/b$ و مقادیر (g, u, v) را بازگردان.

(ج) g را بر y تقسیم کن، $g = qy + t$، که $۰ \leq t < y$.

(د) قرار بده $s = u - qx$.

(ه) قرار بده $u = x$ و $g = y$.

(و) قرار بده $x = s$ و $y = t$.

(ز) به گام ۲ برو.

به زبان کامپیوتری انتخاب خودتان الگوریتم فوق را روی یک کامپیوتر پیاده‌سازی کنید.

از برنامه‌تان برای محاسبه‌ی $g = gcd(a, b)$ و پاسخ‌های صحیح معادله‌ی $au + bv = g$ برای زوج‌های (a, b) زیر استفاده کنید.

(آ) $(۵۲۷, ۱۲۵۸)$

(ب) $(۲۲۸, ۱۰۵۶)$

(ج) $(۱۶۳۹۶۱, ۱۶۷۱۸۱)$

(د) $(۳۸۹۲۳۹۴, ۲۳۹۸۴۷)$

اگر $b = ۰$ باشد چه اتفاقی برای برنامه‌ی شما می‌افتد؟ برنامه را به نحوی اصلاح کنید که هنگام مواجهه با این حالت به درستی عمل کند.

گاهی اوقات مناسب است تا یک جواب $u > ۰$ داشته باشیم. برنامه‌ی خود را به نحوی اصلاح کنید که یک جواب با $u > ۰$ بازگرداند و u کوچکترین مقدار ممکن باشد.[راهنمایی: اگر (u, v) یک جواب باشد، آن‌گاه $(u + b/g, v - a/g)$ نیز یک جواب است.] قسمت (ج) را با برنامه‌ی اصلاح شده انجام دهید.

۱۳.۱ فرض کنید a_1، a_2، ...، a_k اعداد صحیح باشند و $gcd(a_1, a_2, \ldots, a_k) = 1$ یعنی بزرگترین عدد صحیح مثبتی که همه‌ی a_1، a_2، ...، a_k را عاد می‌کند ۱ است. ثابت کنید معادله‌ی

$$a_1u_1 + a_2u_2 + \ldots + a_ku_k = 1$$

در مجموعه‌ی اعداد صحیح دارای جواب u_1، u_2، ...، u_k است. *(راهنمایی. مکرراً الگوریتم توسعه یافته‌ی اقلیدس، قضیه‌ی ۱۴.۱ را به کار برید. شاید اثبات حالت کلی‌تر حکم که در آن $gcd(a_1, a_2, \ldots, a_k)$ می‌تواند بزرگتر از یک باشد.)*

بخش ۱.۳ محاسبات پیمانه‌ای

۱۴.۱ فرض کنید $m \geq 1$ یک عدد صحیح باشد و فرض کنید

$$a_1 \equiv a_2 \pmod{m}, \quad b_1 \equiv b_2 \pmod{m}.$$

ثابت کنید

$$a_1 \pm b_1 \equiv a_2 \pm b_2 \pmod{m}, \quad a_1 \cdot b_1 \equiv a_2 \cdot b_2 \pmod{m}$$

(این قسمت الف گزاره‌ی ۱۸.۱ است.)

۱۵.۱ جدول $\mathbb{Z}/m\mathbb{Z}$ و $(\mathbb{Z}/m\mathbb{Z})^*$ را برای موارد زیر رسم کنید، همان‌طور که ما برای شکل‌های ۳. ۴ و ۵.۱ انجام دادیم.

جدول ضرب و جمع را برای $\mathbb{Z}/3\mathbb{Z}$ بکشید.

جدول ضرب و جمع را برای $\mathbb{Z}/6\mathbb{Z}$ بکشید.

جدول ضرب و جمع را برای گروه یکال‌های $(\mathbb{Z}/9\mathbb{Z})^*$ بکشید.

جدول ضرب و جمع را برای گروه یکال‌های $(\mathbb{Z}/16\mathbb{Z})^*$ بکشید.

۱۶.۱ محاسبات پیمانه‌ای زیر را انجام دهید. در هر حالت، مستطیل را با یک عدد صحیح بین ۰ و $m-۱$، که در آن m پیمانه است، پر کنید.

$$۳۴۷ + ۵۱۳ \equiv \qquad (mod\ ۷۶۳).$$

$$۳۲۷۴ + ۱۲۳۸ + ۷۲۳۱ + ۶۴۳۷ \equiv \qquad (mod\ ۹۲۵۴).$$

$$۳۵۷ \cdot ۸۶۲ \cdot ۱۹۳ \equiv \qquad (mod\ ۹۴۳).$$

$$۵۳۲۷ \cdot ۶۱۳۵ \cdot ۷۱۳۹ \cdot ۲۱۸۷ \cdot ۵۲۱۹ \cdot ۱۸۷۳ \equiv \qquad (mod\ ۸۱۵۷).$$

(راهنمایی: پس از هر ضرب، پیش از انجام ضرب بعدی، به پیمانه‌ی ۸۱۵۷ کاهش دهید.)

$$۱۳۷^{۲} \equiv \qquad (mod\ ۳۲۷).$$

$$۳۷۳^{۶} \equiv \qquad (mod\ ۵۸۱).$$

$$۲۳^{۳} \cdot ۱۹^{۵} \cdot ۱۱^{۴} \equiv \qquad (mod\ ۹۷).$$

۱۷.۱ تمام مقادیر x بین ۰ و $m-۱$ که جواب همنهشتی‌های زیر هستند را بیابید.*(راهنمایی: اگر نمی‌توانید راهی هوشمندانه برای یافتن جواب‌ها بیابید، می‌توانید هر یک از مقادیر $x=۱$، $x=۲$، ...، $x=m-۱$ را جایگذاری کنید و ببینید کدامیک کار می‌کنند.)*

$$x + ۱۷ \equiv ۲۳ \ \ (mod\ ۳۷).$$

$$x + ۴۲ \equiv ۱۹ \ \ (mod\ ۵۱).$$

$$x^{۲} \equiv ۳ \ \ (mod\ ۱۱).$$

$$x^{۲} \equiv ۲ \ \ (mod\ ۱۳).$$

$$x^{۲} \equiv ۱ \ \ (mod\ ۸).$$

$$x^{۳} - x^{۲} + ۲x - ۲ \equiv ۰ \pmod{۱۱}.$$

($x \equiv ۱ \pmod{۵}$ و نیز $x \equiv ۲ \pmod{۷}$.) تمام جواب‌ها به پیمانه‌ی ۳۵ را بیابید، یعنی تمام جواب‌هایی که در $۰ \leq x < ۳۵$ صدق می‌کنند.

۱۸.۱ فرض کنید $g^{a} \equiv ۱ \pmod{m}$ و $g^{b} \equiv ۱ \pmod{m}$. ثابت کنید

$$g^{gcd(a,b)} \equiv ۱ \pmod{m}.$$

۱۹.۱ ثابت کنید اگر $a_{۱}$ و $a_{۲}$ به پیمانه‌ی m یکال باشد، $a_{۱}a_{۲}$ نیز به پیمانه‌ی m یکال است.

۲۰.۱ ثابت کنید m اول است اگر و تنها اگر $\phi(m) = m - ۱$، که در آن ϕ تابع فی اویلر است.

۲۱.۱ فرض کنید $m \in \mathbb{Z}$.

فرض کنید m فرد باشد. کدام اعداد صحیح بین ۱ و $m - ۱$ به پیمانه‌ی m با $۲^{-۱}$ همنهشت هستند؟

به طور کلی‌تر، فرض کنید $m \equiv ۱ \pmod{b}$. کدام یک از اعداد صحیح بین ۱ و $m - ۱$ به پیمانه‌ی m برابر $b^{-۱}$ هستند؟

۲۲.۱ فرض کنید m یک عدد صحیح فرد و a یک عدد صحیح دلخواه باشد. ثابت کنید که $۲m + a^{۲}$ هیچ وقت نمی‌تواند مربع کامل باشد.(راهنمایی: اگر یک عدد مربع کامل باشد، به پیمانه‌ی ۴ چه مقادیری می‌تواند بگیرد؟)

۲۳.۱ یک مقدار ساده‌ی x که دو همنهشتی‌های زیر را به طور همزمان حل می‌کند را بیابید.

$$x \equiv ۳ \pmod{۷}, \qquad x \equiv ۴ \pmod{۹}.$$

(راهنمایی: توجه کنید هر جواب همنهشتی اول به شکل $x = ۳ + ۷y$ است که در آن y یک عدد صحیح است. این مقدار را در همنهشتی دوم جایگزین کرده و همنهشتی را برای y حل کنید؛ سپس از آن برای بدست آوردن x استفاده کنید.)

یک مقدار ساده‌ی x که هر دو همنهشتی

$$x \equiv ۱۳ \pmod{۷۱}, \qquad x \equiv ۴۱ \pmod{۹۷}.$$

یک مقدار ساده‌ی x که هر دو همنهشتی

$$x \equiv ۴ \pmod{۷}, \qquad x \equiv ۵ \pmod{۸}, \qquad x \equiv ۱۱ \pmod{۱۵}.$$

ثابت کنید اگر $gcd(m, n) = ۱$، آن‌گاه زوج همنهشتی‌های

$$x \equiv a \pmod{m}, \qquad x \equiv b \pmod{n}$$

برای هر a و b یک جواب دارد. نشان دهید که شرط $gcd(a, b) = ۱$ ضروری است.

۲۴.۱ فرض کنید N، g و A اعداد صحیح مثبت باشند، (توجه کنید N لزوماً اول نیست.) ثابت کنید الگوریتم زیر، که نسخه‌ی کم جای الگوریتم مربع-و-ضرب توصیف شده در بخش ۱.۳.۲ است، مقدار $g^A \pmod{N}$ را بازمی‌گرداند.(در گام چهار تابع جزء صحیح را با $[x]$ نمایش می‌دهیم.)

Input. Positive integers N, g, and A.
1. Set $a = g$ and $b = 1$.
2. Loop while $A > 0$.
  3. If $A \equiv 1 \pmod 2$, set $b = b \cdot a \pmod N$.
  4. Set $a = a^2 \pmod N$ and $A = \lfloor A/2 \rfloor$.
  5. If $A > 0$, continue with loop at Step **2**.
6. Return the number b, which equals $g^A \pmod N$.

۲۵.۱ با استفاده از الگوریتم مربع- و- ضرب توصیف شده در بخش ۱.۳۰، یا نسخه‌ی کارآتر تمرین ۲۴.۱، توان‌های زیر را محاسبه کنید.

$$۱۷^{۱۸۳} \pmod{۲۵۶}.$$

$$۲^{۴۷۷} \pmod{۱۰۰۰}.$$

$$۱۱^{۵۰۷} \pmod{۱۲۳۷}.$$

بخش ۱.۴ اعداد اول، یکتایی تجزیه و میدان‌های متناهی

۲۶.۱ فرض کنید $\{p_۱, p_۲, \ldots, p_r\}$ مجموعه‌ای از اعداد اول باشند، و قرار دهید

$$N = p_۱ p_۲ \ldots p_r + ۱.$$

ثابت کنید که N توسط عددی اول که در مجموعه نیست عاد می‌شود. از این حقیقت استفاده کنید تا نتیجه بگیرید باید بینهایت عدد اول وجود داشته باشد.(این اثبات نامتناهی بودن اعداد اول در اصول اقلیدس ظاهر می‌شود. اعداد اول برای صدها سال مورد مطالعه بوده‌اند.)

۲۷.۱ بدون استفاده از این حقیقت که هر عدد صحیح تجزیه‌ای یکتا به اعداد اول دارد، ثابت کنید اگر $gcd(a, b) = ۱$ و $a|bc$، آن‌گاه $a|c$. (راهنمایی: از این حقیقت استفاده کنید که می‌توان جوابی برای $au + bv = ۱$ یافت.)

۲۸.۱ مقادیر ord_p زیر را محاسبه کنید:

$$ord_۲(۲۸۱۶).$$

$$ord_۷(۲۲۲۲۵۷۴۴۸۷).$$

برای هر $p = ۳, ۵, ۷, ۱۱$، $ord_p(۴۶۳۷۵)$

۲۹.۱ فرض کنید p یک عدد اول باشد. ثابت کنید ord_p خواص زیر را دارد.

$ord_p(ab) = ord_p(a) + ord_p(b)$. *(بنابراین ord_p شبیه تابع لگاریتم است، زیرا ضرب را به جمع تبدیل می‌کند!)*

$ord_p(a+b) \geq min\{ord_p(a), ord_p(b)\}$.

اگر $ord_p(a) \neq ord_p(b)$، آن‌گاه $ord_p(a+b) = min\{ord_p(a), ord_p(b)\}$.

تابعی که در خواص (۱) و (۲) صدق می‌کند را یک ارزیاب می‌گوییم.

بخش ۱.۵ توان‌ها و ریشه‌های اولیه در میدان‌های متناهی

۳۰.۱ برای هر یک از اعداد اول p و اعداد aی زیر، به دو طریق $a^{-1} \pmod p$ را محاسبه کنید: *(الف)* از الگوریتم توسعه یافته‌ی اقلیدس استفاده کنید. *(ب)* از قضیه‌ی کوچک فرما و الگوریتم به توان رساندن سریع استفاده کنید. *(به تمرین ۲۸.۱ مراجعه کنید.)*

$p = ۴۷$ و $a = ۱۱$.

$p = ۵۸۷$ و $a = ۳۴۵$.

$p = ۱۰۴۸۰۱$ و $a = ۷۸۴۶۷$.

۳۱.۱ فرض کنید p یک عدد اول باشد و q عددی اول باشد که $p - ۱$ را عاد می‌کند.

فرض کنید $a \in \mathbb{F}_p^*$ و $b = a^{(p-۱)/q}$. ثابت کنید یا $b = ۱$ یا مرتبه‌ی b برابر q است.*(به یاد آورید که مرتبه‌ی b کوچکترین عدد صحیح $k \geq ۱$ است به نحوی که در $\mathbb{F}_p^*$، $b^k = ۱$. راهنمایی: از گزاره‌ی ۴۰.۱ استفاده کنید.)*

فرض کنید می‌خواهیم عنصری از مرتبه‌ی q در $\mathbb{F}_p^*$ بیابیم. با استفاده از قسمت *(الف)* می‌توانیم به طور تصادفی یک مقدار $a \in \mathbb{F}_p^*$ را انتخاب کنیم و بررسی کنیم آیا $b = a^{(p-۱)/q}$ در $b \neq ۱$ صدق می‌کند؟ احتمال موفقیت ما چقدر است؟ به عبارت دیگر مقدار

$$\frac{\#\{a \in \mathbb{F}_p^* : \ a^{(p-۱)/q} \neq ۱\}}{\#\mathbb{F}_p^*}$$

را محاسبه کنید.(راهنمایی: از قضیه‌ی ۴۱.۱ استفاده کنید.)

۳۲.۱ به یاد آورید که g را یک ریشه‌ی اولیه به پیمانه‌ی p می‌خوانیم اگر توان‌های g همه‌ی اعضای ناصفر $\mathbb{F}_p$ را ایجاد کنند.

برای کدام یک از اعداد اول زیر ۲ یک ریشه‌ی اولیه به پیمانه‌ی p است؟

$$(i) \ \ p = ۷ \quad (ii) p = ۱۳ \quad (iii) p = ۱۹ \quad (iv) p = ۲۳$$

برای کدام یک از اعداد اول زیر ۳ یک ریشه‌ی اولیه به پیمانه‌ی p است؟

$$(i) \ \ p = ۵ \quad (ii) p = ۷ \quad (iii) p = ۱۱ \quad (iv) p = ۱۷$$

برای هر یک از اعداد اول زیر یک ریشه‌ی اولیه بیابید.

$$(i) \ \ p = ۲۳ \quad (ii) p = ۲۹ \quad (iii) p = ۴۱ \quad (iv) p = ۴۳$$

همه‌ی ریشه‌های اولیه به پیمانه‌ی ۱۱ را بیابید. تحقیق کنید $\phi(۱۰)$ تا از آن‌ها وجود دارد، همان‌طور که در تذکر ۴۳.۱ ادعا شد.

یک برنامه‌ی کامپیوتری برای بررسی ریشه‌های اولیه بنویسید و از آن برای یافتن همه‌ی ریشه‌های اولیه به پیمانه‌ی ۲۲۹ استفاده کنید. تحقیق کنید دقیقاً $\phi(۲۲۹)$ تا از آن‌ها وجود دارد.

از برنامه‌ی خود در قسمت *(ه)* برای یافتن تمام اعداد اول کوچکتر از ۱۰۰ که ۲ ریشه‌ی اولیه‌ی آن‌ها است استفاده کنید.

تمرین قبل را برای یافتن تمام اعداد اول کوچکتر از ۱۰۰ که ۳ ریشه‌ی اولیه‌ی آن‌ها است استفاده کنید. همچنین برای یافتن تمام اعداد اولی که ۴ ریشه‌ی اولیه‌ی آن‌ها است.

۳۳.۱ فرض کنید p یک عدد اول باشد به نحوی که $q = \frac{1}{2}(p-1)$ نیز اول است. فرض کنید g عددی صحیح است که در

$$g \not\equiv \pm 1 \pmod p, \quad g^q \not\equiv 1 \pmod p$$

صدق می‌کند. ثابت کنید g یک ریشه‌ی اولیه به پیمانه‌ی p است.

۳۴.۱ این تمرین مطالعه‌ی مربع‌ها و مجذورها به پیمانه‌ی p را آغاز می‌کند.

فرض کنید p یک عدد اول فرد باشد و b عددی صحیح باشد که $p \nmid b$. ثابت کنید یا b دو جذر به پیمانه‌ی p دارد یا b هیچ جذری به پیمانه‌ی p ندارد. به عبارت دیگر، ثابت کنید همنهشتی

$$X^2 \equiv b \pmod p$$

در $\mathbb{Z}/p\mathbb{Z}$ یا دو جواب دارد یا هیچ جوابی ندارد.(برای $p = 2$ چه اتفاقی می‌افتد؟ اگر $p|b$ چه اتفاقی می‌افتد؟)

برای هر یک از مقادیر p و b زیر، همه‌ی جذرهای b به پیمانه‌ی p را بیابید.

$$(i)\ \ (p,b) = (7,2) \qquad (ii)\ \ (p,b) = (11,5)$$

$$(iii)\ \ (p,b) = (11,7) \quad (iv)\ \ (p,b) = (37,3)$$

۲۹ به پیمانه‌ی ۳۵ چند جذر دارد؟ چرا با ادعای (الف) تناقض دارد؟

فرض کنید p یک عدد اول فرد و g یک ریشه‌ی اولیه به پیمانه‌ی p باشد. در این صورت هر عدد a با توانی از g به پیمانه‌ی p برابر است.، برای مثال $a = g^k \pmod p$. ثابت کنید a به پیمانه‌ی p جذر دارد اگر و تنها اگر k زوج باشد.

۳۵.۱ فرض کنید $p \geq 3$ یک عدد اول باشد و همنهشتی

$$X^{۲} \equiv b \pmod p$$

یک جواب داشته باشد.

ثابت کنید برای هر توان $e \geq ۱$ همنهشتی

$$X^{۲} \equiv b \pmod{p^e} \qquad (۱۶.۱)$$

یک جواب دارد. *(راهنمایی: از استقراء روی e استفاده کنید. با اصلاح مناسب یک جواب به پیمانه‌ی p^e، یک جواب به پیمانه‌ی $p^{e+۱}$ بسازید.)*

فرض کنید $X = \alpha$ جوابی برای $X^{۲} \equiv b \pmod p$ باشد. ثابت کنید در قسمت *(الف)* می‌توانیم یک جواب $X = \beta$ برای $X^{۲} \equiv b \pmod{p^e}$ بیابیم که در $\alpha \equiv \beta \pmod{p^e}$ نیز صدق می‌کند.

فرض کنید β و β' جواب‌هایی مثل قسمت *(ب)* باشند. ثابت کنید $\beta \equiv \beta' \pmod p$.

از تمرین ۳۴.۱ استفاده کنید تا نشان دهید همنهشتی(۱۶.۱) به پیمانه‌ی p^e یا دو جواب دارد یا هیچ جوابی ندارد.

۳۶.۱ برای هر عدد اول $۳ \leq p < ۲۰$ مقدار

$$۲^{(p-۱)/۲} \pmod p$$

را محاسبه کنید. در خصوص مقادیر ممکن برای $۲^{(p-۱)/۲} \pmod p$ در حالتی که p اول است یک حدس بزنید و ثابت کنید حدستان درست است.

بخش ۱.۶ رمزنگاری با دست

۳۷.۱ در خصوص یکی از موضوعات زیر ۲ تا ۵ صفحه مطلب بنویسید، هم شامل اطلاعات رمزنگاری هم رخدادهای موجود در تاریخ آن‌ها:

رمزنگاری در دنیای اعراب تا قرن ۱.

رمزنگاری اروپایی در قرن ۱۵ و اوایل قرن ۱.

رمزنگاری و تحلیل رمز در انگلستان ملکه الیزابت.

رمزنگاری و تحلیل رمز در قرن ۱.

رمزنگاری و تحلیل رمز طی جنگ جهانی اول.

رمزنگاری و تحلیل رمز طی جنگ جهانی دوم.

(اکثر این موضوعات برای یک مقاله مختصر وسیع هستند، بنابراین باید یک جنبه خاص را در نظر بگیرید تا تمرکز کنید.)

۳۸.۱ یک رمز هم‌آهنگ یک رمز جانشانی است که در آن ممکن است برای یک حرف متن ساده بیش از یک نماد متن رمز وجود داشته باشد. در اینجا مثالی از یک رمز هماهنگ آمده است، که در آن حروف متداول امکان‌های زیادی دارند.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
!	4	#	$	1	%	&	*	(	)	3	2	=	+	[	9	]	{	}	:	;	7	<	>	5	?
♡	∘	⋆	ℵ	6	↗	▷	♢	∧			↘	∆	∇	8	♣		Ω	∨	⊗	♠				♭	
Θ				∞		⇑		♮					•	⊙				◁	⊕	⇐					
↙				⇓														⇒	↖						

پیام زیر را رمزگشایی کنید.

(% ∆ ♠ ⇒ ♮ # 4 ∞ : ♢ 6 ↗ ⊙ [ℵ 8 % 2 [7 ⇓ ♣ ↘ ♡ 5 ⊙ ∇

۳۹.۱ یک رمز ترانهشی که در آن حروف متن ساده، همان می‌مانند، اما ترتیب آن‌ها تغییر می‌کند. در اینجا مثالی وجود دارد که در یک زمان، پیام با بلوک‌های ۲۵ حرفی رمز می‌شود. ۲۵ حرف داده

شده را گرفته و با نوشتن افقی پیام در خطوط آن را در یک بلوک قرار دهید. برای مثال، ۲۵ حرف اول پیام

Now is the time for all good men to come to the eid...

به شکل

N	O	W	I	S
T	H	E	T	I
M	E	F	O	R
A	L	L	G	O
O	D	M	E	N

نوشته می‌شود. حال متن رمز خواندن حروف از ستون‌ها تشکیل می‌شود، که متن رمز

NTMAO OHELD WEFLM ITOGE SIRON

را می‌دهد.

از رمز ترانهشی برای رمز کردن ۲۵ حرف اول پیام

Four score and seven years ago our fathers...

استفاده کنید.

متن زیر با استفاده از رمز ترانهشی رمز شده است. آن را رمزگشایی کنید.

WNOOA HTUFN EHRHE NESUV ICEME

نسخه‌های مختلفی از این نوع رمز وجود دارند. می‌توانیم به جای یک مربع حروف را روی یک مستطیل بچینیم، و از الگوهای مختلف برای قرار دادن حروف در مستطیل و بازخوانی

WHNCE STRHT TEOOH ALBAT DETET SADHE
LEELL QSFMU EEEAT VNLRI ATUDR HTEEA

آن‌ها استفاده کنیم. سعی کنید متن رمز زیر را رمزگشایی کنید، که در آن حروف به طور افقی در مستطیلی چیده شده و سپس با عمودی خواندن ستون‌ها شناسایی می‌شوند.

(برای راحتی، متن رمز را در بلوک‌های ۵ حرفی نوشته‌ایم، اما این دلیل نمی‌شود که مستطیل ضلعی به اندازه‌ی ۵ داشته باشد.)

بخش ۱.۷ رمزهای متقارن و رمزهای نامتقارن

۴۰.۱ با استفاده از طرح کد کردن اسکی که در جدول ۷.۱ داده شده عبارت زیر را کد کنید(شامل حروف بزرگ، فاصله گذاری و نقطه گذاری).

$$Bad\ day, Dad.$$

۴۱.۱ رمز آفین با کلید $k = (k_۱, k_۲)$ که توابع رمز کردن و رمزگشایی آن با (۱۳.۱) در صفحه‌ی ؟؟؟ داده شده است را در نظر بگیرید.

فرض کنید $p = ۵۴۱$ و کلید $k = (۳۴, ۷۱)$ باشد. پیام $m = ۲۰۴$ را رمز کنید. متن رمز $c = ۴۳۱$ را رمزگشایی کنید.

با فرض اینکه p در اختیار عموم است، توضیح دهید چرا رمز آفین در مقابل حمله متن ساده انتخابی آسیب پذیر است. (خاصیت ۴ در صفحه؟؟؟ را ببینید.) برای بازیابی کلید خصوصی حدوداً چند زوج متن ساده/متن رمز نیاز است؟

آلیس و باب تصمیم دارند تا برای رمز آفین خود از عدد اول $p = ۶۰۱$ استفاده کنند. مقدار p در اختیار عموم قرار دارد، و اِو جلوی متون رمزی $c_۱ = ۳۲۴$ و $c_۲ = ۳۸۱$ را گرفته و نیز موفق به کشف متون ساده متناظر $m_۱ = ۳۸۷$ و $m_۲ = ۴۹۱$ می‌شود. کلید خصوصی را تعیین کرده و با استفاده از آن پیام $m_۳ = ۱۷۳$ را رمز کنید.

حال فرض کنید p در اختیار عموم نیست. آیا رمز آفین هنوز هم در مقابل حمله‌ی متن ساده انتخابی آسیب پذیر است؟اگر چنین است، تقریباً چه تعداد زوج متن ساده/متن رمزی برای بازیابی کلید خصوصی احتیاج است؟

۴۲.۱ رمز هیل که با (۱۳.۱) تعریف شد را در نظر بگیرید،

$$e_k(m) \equiv k_1 \cdot m + k_2 \pmod p, d_k(c) \equiv k_1^{-1} \cdot (c - k_2) \pmod p,$$

که در آن m، c و k_2 بردارهای ستونی از بعد n و k_1 یک ماتریس n در n است.

ما از رمز هیل با $p = 7$ و کلید $k_1 =???$ و $k_2 =??$ استفاده می‌کنیم.

(آ) پیام $m =???$ را رمز کنید.

(ب) ماتریس k_1^{-1} استفاده شده برای رمزگشایی چیست؟

(ج) پیام $c =???$ را رمزگشایی کنید.

توضیح دهید چرا رمز هیل در مقابل حمله متن ساده انتخابی آسیب پذیر است.

زوج‌های متن ساده/متن رمزی زیر به ازای $p = 11$ با استفاده از رمز هیل بدست آمده‌اند. کلیدهای k_1 و k_2 را بیابید.

????????????????

توضیح دهید چگونه یک متن جانشانی ساده که شامل یک جایگشت از الفبا است را می‌توان حالت خاصی از یک رمز هیل در نظر گرفت.

۴۳.۱ فرض کنید N یک عدد صحیح بزرگ باشد و $\mathcal{K} = \mathcal{M} = \mathcal{C} = \mathbb{Z}/N\mathbb{Z}$. برای هر یک از توابع

$$e : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$$

که در قسمت‌های *(الف)*، *(ب)* و *(ج)* آمده‌اند به سئوالات زیر پاسخ دهید:

- آیا e یک تابع رمز کردن است؟
- اگر e یک تابع رمز کردن است، تابع رمزگشاییd متناظر آن چیست؟
- اگر e یک تابع رمز کردن نیست، آیا می‌توانید با استفاده از مجموعه‌ای کوچکتر، در عین حال بزرگ، از کلیدها از آن یک تابع رمز ساخت؟

$$e_k(m) \equiv k - m \pmod N.$$

$$e_k(m) \equiv k \cdot m \pmod N.$$

$$e_k(m) \equiv (k+m)^{۲} \pmod N.$$

۴۴.۱ عدد دودویی ۱۱۰۱۰۱۱۰۰۱۰۱ را به یک عدد دهدهی بین ۰ و $۲^{۱۲} - ۱$ تبدیل کنید.

عدد صحیح $m =$ ۳۷۸۵۳ را به یک عدد دودویی تبدیل کنید.

عدد صحیح $m =$ ۹۴۸۷۴۲۸ را به یک عدد دودویی تبدیل کنید.

از یای مانعة الجمع *(XOR)* برای جمع رشته‌های دودویی ۱۰۰۱۱۰۱۰ $\oplus$ ۱۱۰۰۱۰۱۰ استفاده کنید.

اعداد دهدهی ۸۷۳۴ و ۵۱۷۷ را به اعداد دودویی تبدیل کنید، با استفاده از XOR ان‌ها را ترکیب کنید، و نتیجه را به یک عدد دهدهی تبدیل کنید.

۴۵.۱ آلیس و باب یک فضای کلید $\mathcal{K}$ شامل $۲^{۵۶}$ کلید را انتخاب می‌کنند. اِو یک کامپیوتر خاص که می‌تواند در هر ثانیه ۱۰,۰۰۰,۰۰۰,۰۰۰ کلید بررسی کند را می‌سازد.

چند روز طول می‌کشد تا اِو نصف کلیدهای موجود در $\mathcal{K}$ را بررسی کند؟

آلیس و باب فضای کلیدشان را با مجموعه‌ای بزرگتر شامل ۲^B کلید متفاوت جایگزین می‌کنند. برای اینکه کامپیوتر اِو برای بررسی نصف کلیدها ۱۰۰ سال لازم داشته باشد، آلیس و باب B را چقدر بزرگ انتخاب کنند؟(از این تقریب استفاده کنید که هر سال حدود ۳۶۵.۲۵ روز دارد.) برای سالها دولت ایالات متحده یک رمز متقارن معروف به DES که از کلیدهای ۵۶ بیتی استفاده می‌کند را توصیه می‌کردند. در ۱۹۹، مردم کامپیوترهای تک منظوره‌ای ساختند که نشان می‌داد این ۵۶ بیت امنیت کافی فراهم نمی‌کند. یک رمز متقارن معروف به AES، با کلیدهای ۱۲۸ بیتی، برای جایگزینی DES معرفی شدند. برای اطلاعات بیشتر در خصوص DESو AES به بخش ۸.۱۰ مراجعه کنید.

۴۶.۱ توضیح دهید چرا رمز

$$e_k(m) = k \oplus m, \quad d_k(c) = k \oplus c$$

که باXOR رشته‌های بیتی تعریف شده است در مقابل حمله‌ی متن ساده‌ی انتخابی امن نیست. با یافتن کلید خصوصی مورد استفاده برای رمز کردن متن رمزی ۱۶ بیتی $c =$ ۱۰۰۱۰۱۰۰۰۱۰۱۰۱۱۱ حمله خود را توضیح دهید، در حالیکه می‌دانید متن ساده متناظر عبارتست از $m =$ ۰۰۱۰۰۱۰۰۰۰۱۰۱۱۰۰.

۴۷.۱ آلیس و باب به صورت زیر یک رمز متقارن می‌سازند. کلید خصوصی آن‌ها k یک عدد صحیح بزرگ است و پیام‌های آن‌ها (متون ساده) اعداد صحیح d- رقمی هستند

$$\mathcal{M} = \{m \in \mathbb{Z} : \quad ۰ \leq m < ۱۰^d\}.$$

برای رمز کردن یک پیام، آلیس $\sqrt{k}$ را تا d رقم اعشار محاسبه می‌کند، قسمت سمت چپ اعشار را به دور ریخته و d رقم باقیمانده را نگه می‌دارد. فرض کنید α این عدد d-رقمی باشد. (برای مثال اگر $k =$ ۲۳ و $d =$ ۶ آن‌گاه $\sqrt{۸۷} =$ ۹٫۳۲۷۳۷۹۰۵... و $\alpha =$ ۳۲۷۳۷۹.)

آلیس پیام m را به صورت

$$c \equiv m + \alpha \pmod{۱۰^d}$$

رمز می‌کند. از آن‌جا که باب k را می‌داند، او نیز می‌تواند α را بیابد، و سپس با استفاده از $m \equiv c - \alpha \pmod{۱۰^d}$، c را رمزگشایی کند.)

آلیس و باب کلید مخفی $k = ۱۱$ را انتخاب کرده و از آن برای رمز کردن اعداد صحیح استفاده می‌کنند (یعنی $d = ۶$). باب می‌خواهد پیام $m = ۳۲۸۹۷۳$ را برای آلیس ارسال کند. متن رمزی که باید بفرستد چیست؟

آلیس و باب کلید مخفی $k = ۲۳$ را انتخاب کرده و از آن برای رمز کردن اعداد ۸ رقمی استفاده می‌کنند. آلیس متن رمزی $c = ۸۷۱۸۳۹۰۳$ را دریافت می‌کند. پیام ساده‌ی m چیست؟

نشان دهید عدد α مورد استفاده در رمز کردن و رمز گشایی با فرمول

$$\alpha = \lfloor ۱۰^d(\sqrt{k} - \lfloor \sqrt{k} \rfloor) \rfloor$$

داده می‌شود که در آن $\lfloor t \rfloor$ نشان‌دهنده‌ی بزرگترین عدد صحیح کوچکتر از t است.

(مسئله‌ی مشکل) اگر اِو یک زوج متن ساده/ متن رمز (m, c) را بدزدد، واضح است که می‌تواند عدد α را بیابید، زیرا $\alpha \equiv c - m \pmod{۱۰^d}$. اگر ۱۰^d نسبت به k بزرگ باشد، آیا می‌تواند k را نیز بدست آورد؟ این ممکن است مفید باشد، برای مثال، اگر آلیس و باب از برخی دیگر از ارقام $\sqrt{k}$ برای پیام‌های بعدی استفاده کنند.

۴۸.۱ آلیس و باب از یک سیستم رمزنگاری استفاده می‌کنند که در آن کلید خصوصی‌شان یک عدد اول بزرگ k متون ساده و متون رمزی آن‌ها اعداد صحیح هستند. باب با محاسبه‌ی $c = km$ پیام m را رمز می‌کند. باب جلوی متون رمزی زیر را می‌گیرد

$$c_۱ = ۱۲۸۴۹۲۱۷۰۴۵۰۰۶۲۲۲, \qquad c_۲ = ۶۴۸۵۸۸۰۴۴۳۶۶۶۲۲۲.$$

از روش gcd که در بخش ۱.۷.۴ توضیح داده شد برای یافتن کلید خصوصی آلیس و باب استفاده کنید.