رمزنگاری سنتی

مدرس:

محمد غلامی

دکترای ریاضی، دانشیار دانشگاه شهرکرد

مقدمه. در این فصل، یک مقدمه کلی درباره رمزنگاری و شکستن یک سیستم رمزنگاری را مورد بحث و بررسی قرار خواهیم داد. چندین سیستم ساده را معرفی کرده و نحوه شکستن آنها را توضیح خواهیم داد. در این راستا، چندین روش ریاضی را که در ادامه مطالب مورد استفاده قرار می گیرد، معرفی خواهیم کرد.

فهرست مطالب

برخی سیستم های رمزنگاری ساده

- رمز شیفتی
- رمز جانشینی
- رمز آفین
- رمز ویجینر
- رمز هیل
- رمز جایگشتی
- رمز دنباله ای

رمز شکنی سیستم های رمزنگاری سنتی

فصل ۱. تاریخچه

در اینجا برخی تعاریف و مفاهیم مورد نیاز در بخش های دیگر را می آوریم. برخی از حوادث در تاریخ رمز به صورت زیر می باشند:

رمزنگاری سابقه‌ای طولانی و جذاب دارد، که اولین کاربرد آن به ۴۰۰۰ سال پیش در مصر باستان باز می‌گردد.

در ۱۳۰۰ میلادی ابن خلدون جهت استفاده اداره مالیات و ارتش برای ساده نویسی و پنهان کاری از نوعی رمز استفاده می کرد.

بین سالهای ۱۹۳۳ تا ۱۹۴۵ رمز انگما که به آلمان برده شده بود، تکامل پیدا کرد و مورد استفاده آلمان ها قرار گرفت. این رمز توسط یک ریاضیدان لهستانی به نام مارین رجوسکی شکسته شد.

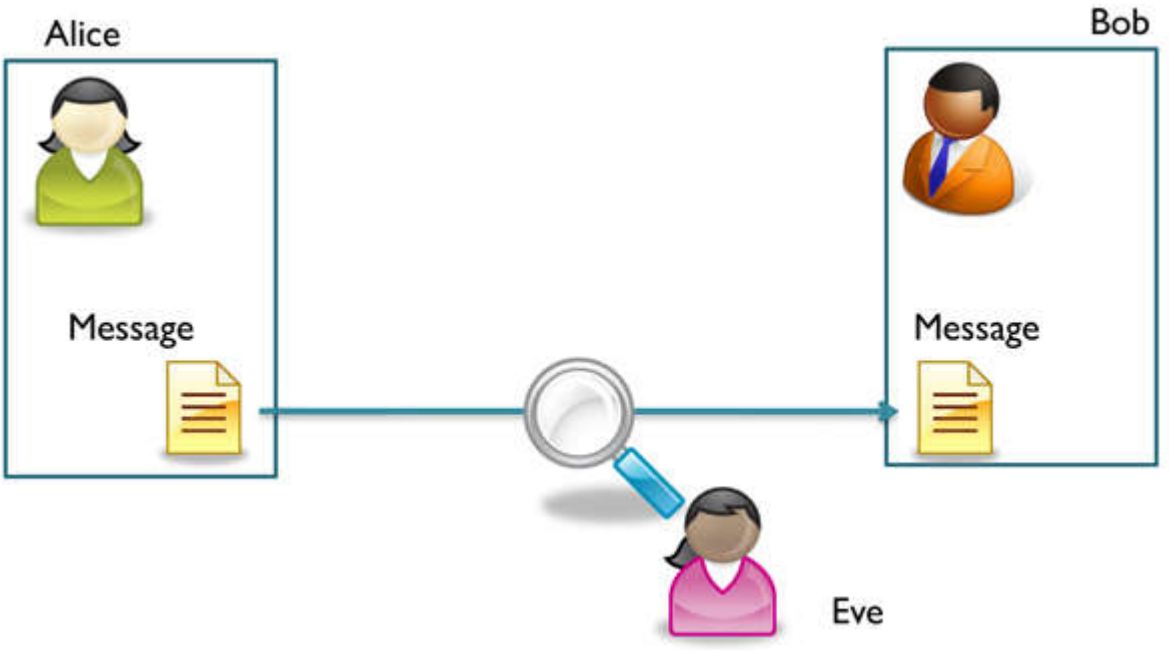
برای اولین بار در سال ۱۹۹۱ رمز کوانتومی توسط بنتووبراساد مطرح شد. آنها از یک فوتون جهت انتقال کلید استفاده کردند. در این رمزکننده، گیرنده و فرستنده باید دارای کابل فیبر نوری باشند.

برخی تعاریف مقدماتی

- **رمزنگاری**: دانش (هنری) است که به بررسی و شناختِ اصول و روش‌های انتقال یا ذخیره اطلاعات به صورت امن می‌پردازد (حتی اگر مسیر انتقال اطلاعات و کانال‌های ارتباطی یا محل ذخیره اطلاعات ناامن باشند). معادل رمزنگاری در زبان انگلیسی کلمه cryptography است، که برگرفته از لغات یونانی kryptos به مفهوم «**محرمانه**» و graphien به معنای «**نوشتن**» است.

- **تحلیل رمز** (cryptanalysis) یا شکستن رمز، به کلیه اقدامات مبتنی بر اصول ریاضی و علمی اطلاق می‌گردد که هدف آن از بین بردن امنیت رمزنگاری و در نهایت بازکردن رمز و دستیابی به اطلاعات اصلی باشد.
- **رمزشناسی** (cryptology): رمزنگاری + تحلیل رمز

رمزنگاری



اصول ششگانه کرشُهف

- سیستم رمزنگاری اگر نه به لحاظ تئوری که در عمل غیرقابل شکست باشد.
- سیستم رمز نگاری باید هیچ نکته پنهان و محرمانه‌ای نداشته باشد. بلکه تنها چیزی که سری است کلید رمز است.
- کلید رمز باید به گونه‌ای قابل انتخاب باشد که اولاً بتوان براحتی آن را عوض کرد و ثانیاً بتوان آنرا به خاطر سپرد و نیازی به یاداشت کردن کلید رمز نباشد.
- متون رمز نگاری باید از طریق خطوط تلگراف قابل مخابره باشند.
- دستگاه رمزنگاری یا اسناد رمز شده باید توسط یک نفر قابل حمل و نقل باشد.
- سیستم رمزنگاری باید به سهولت قابل راه‌اندازی باشد.

برخی مفاهیم اصلی

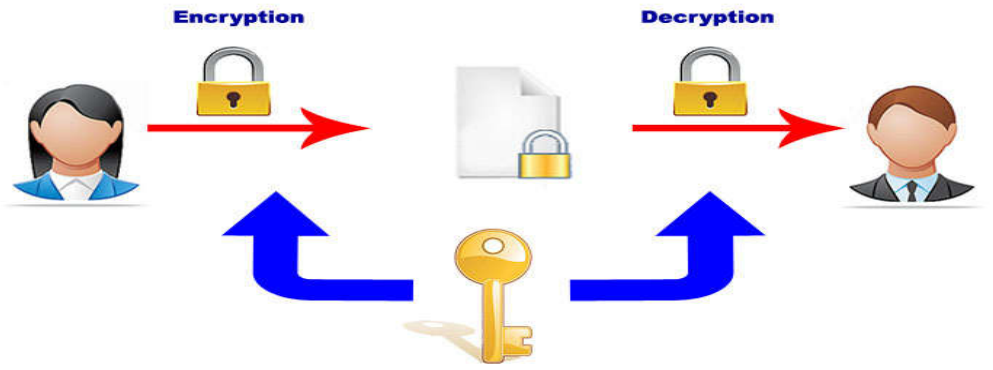
- متن اصلی (Plaintext): پیام خام رمزنشده که می خواهیم آن را ارسال کنیم.
- متن رمزشده (Ciphertext): متن رمزی که برای دیگران بجز افراد مجاز نامفهوم است.
- عمل رمزنگاری (Encryption-Encipher): روند تبدیل متن اصلی به متن رمزشده با استفاده از الگوریتم رمزنگاری
- عمل رمزگشایی (Decryption-Decipher): تبدیل معکوس رمزنگاری با استفاده از الگوریتم رمزگشایی
- کلید (Key): مقداری که برای انجام عملیات بالا از آن استفاده می شود.

اهداف رمزنگاری

- ❑ محرمانگی (Confidentiality): اطمینان از این که تنها شخص مورد نظر (شخصی که دارای کلید است) قادر به استخراج متن ساده از روی متن رمز شده باشد.
- ❑ صحت (Integrity): اطمینان از این که پیامی که به دست گیرنده می رسد، دقیقا همان پیامی باشد که ارسال شده است.
- ❑ احراز هویت (Authentication): به معنای تشخیص و ایجاد اطمینان از هویت ارسال‌کننده اطلاعات و عدم امکان جعل هویت افراد می‌باشد.
- ❑ عدم انکار (Nonrepudiation): به این معنی است که ارسال‌کننده اطلاعات نتواند در آینده ارسال آن را انکار یا مفاد آن را تکذیب نماید.

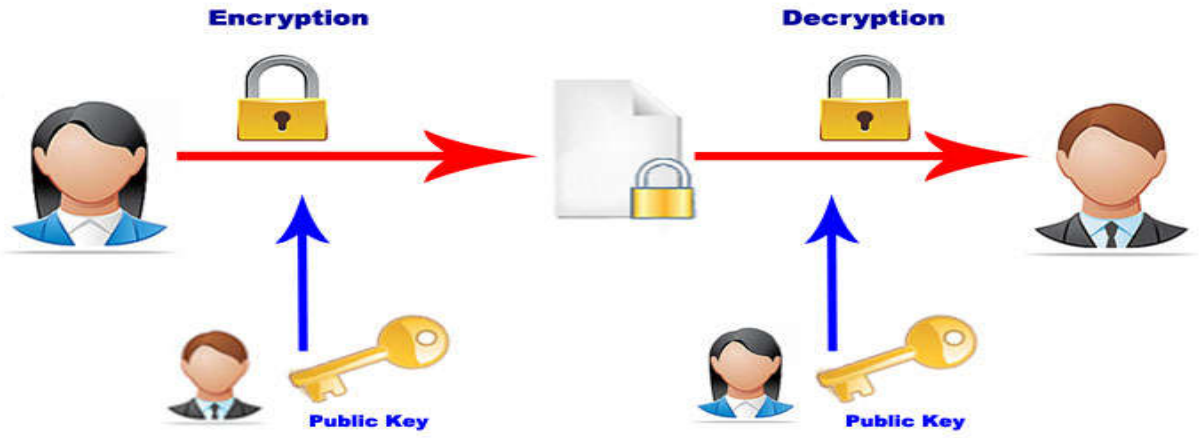
رمزنگاری (کلید) متقارن Symmetric (secret) Key

- رمزنگاری متقارن: از یک کلید برای رمزگذاری و رمزگشایی استفاده می شود یا به دست آوردن کلید رمزگشایی از کلید رمزگذاری ساده است.
- آلیس و باب یک کلید K را از طریق یک کانال امن به اشتراک می گذارند.
- رمزگذاری: متن ساده با استفاده از کلید K رمزگذاری می شود.
- رمزگشایی: متن رمزشده با همان کلید K رمزگشایی می شود.



رمزنگاری کلید عمومی Public Key Encryption

رمزنگاری کلید عمومی یا **رمزنگاری نامتقارن** روشی از رمزنگاری است که کلید مورد استفاده برای رمزگذاری با کلید مربوط برای رمزگشایی با هم متفاوت است



- باب دو کلید K_{eb}(کلید عمومی) و K_{db} (کلید خصوصی) تولید می کند، به گونه ای که به دست آوردن کلید خصوصی از روی کلید عمومی از نظر محاسباتی امکان پذیر نباشد.
- باب کلید K_{eb} را به عنوان کلید عمومی اعلام می کند.
- آلیس متن ساده P را با استفاده از کلید عمومی باب به صورت $C = E(K_{eb}, P)$ رمزگذاری می کند و آن را برای باب ارسال می کند.
- باب متن رمزشده C را با استفاده از کلید K_{db} (کلید خصوصی باب) به صورت $P = D(K_{db}, C)$ رمزگشایی کرده و متن ساده P را به دست می آورد.

سیستم رمزنگاری

تعریف. یک سیستم رمزنگاری شامل یک پنج تایی به صورت (P,C,K,E,D) است که شرایط زیر برقرار باشد:

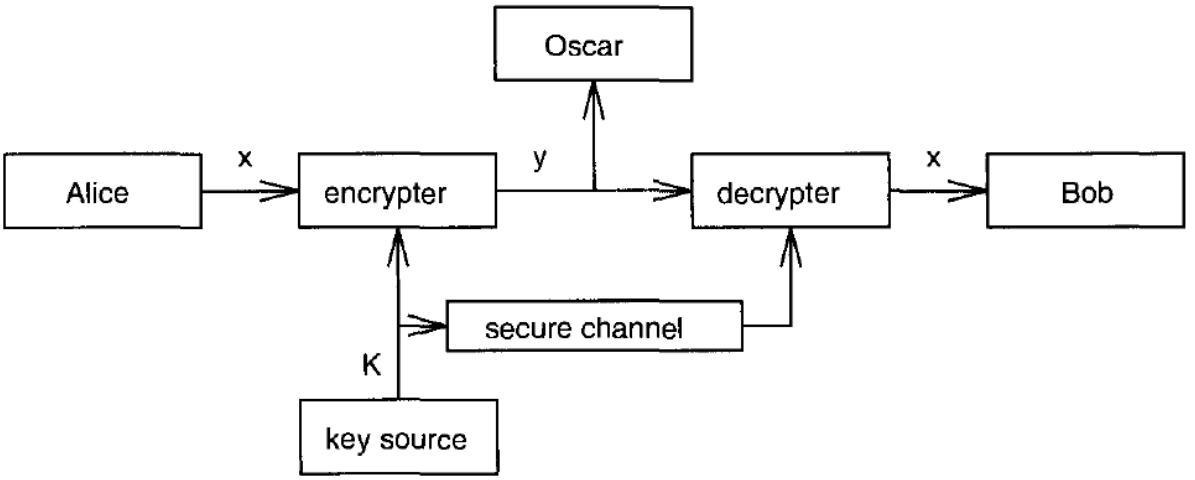
P یک مجموعه متناهی از متن های ساده ممکن است.

C یک مجموعه متناهی از متن های رمزشده ممکن است.

K فضای کلید بوده که مجموعه متناهی از کلیدهای ممکن است.

برای هر $k \in K$ یک قاعده رمزنگاری $e_k \in E$ و یک قاعده رمزگشایی $d_k \in D$ وجود دارد به طوری که $e_k : P \to C$ و $d_k : C \to P$ توابعی هستند که $d_k(e_k(x)) = x$ برای هر $x \in P$.

شمای کلی یک سیستم رمزنگاری



پروتکل سیستم رمزنگاری

آلیس و باب پروتکل زیر را برای یک سیستم رمزنگاری مشخص به کار می برند:

- ابتدا آنها یک کلید $k \in K$ را به صورت تصادفی انتخاب کرده و در جایی که از دید اسکار مخفی باشد (کانال امن)، این کلید را مابین خود مبادله می کنند.
- فرض کنید پیام به صورت یک رشته $x = x_1 \ldots x_n$ باشد که در آن $n \geq 1$ و برای هر $1 \leq i \leq n$، $x_i \in P$ یک سمبل از متن ساده است.
- هر $x_i \in P$ توسط آلیس با استفاده از کلید k به صورت $y_i = e_k(x_i)$ رمز شده و رشته متن رمزشده $y = y_1 \ldots y_n$ برای باب ارسال می شود.
- باب به محض دریافت رشته y، با استفاده از تابع رمزگشای d_k، رشته متن ساده $x = x_1 \ldots x_n$ را به صورت $x_i = d_k(y_i)$، $1 \leq i \leq n$ ، به دست می آورد.

دو نکته مهم

۱. برای هر $k \in K$، تابع e_k باید یک تابع یک به یک باشد، زیرا در غیر این صورت اگر $y = e_k(x_1) = e_k(x_2)$ برای متن های ساده متفاوت $x_1 \neq x_2$، آنگاه باب هیچ روشی برای این که y می بایست به x_1 یا x_2 رمزگشایی گردد، ندارد.

۲. در حالتی که $P = C$، یعنی فضای متن ساده و متن رمزشده یکسان باشد، هر تابع رمزنگاری یک تابع جایگشتی بوده که عناصر مجموعه متن ساده را با یکدیگر جابجا می کند.

خاصیت همنهشتی

فرض کنید a,b دو عدد صحیح بوده و m یک عدد صحیح مثبت باشد. در این صورت، می نویسیم $a \equiv b \pmod{n}$ هرگاه $b-a$ بر m بخش پذیر باشد. در این حالت گوییم a و b همنهشت با یکدیگر در پیمانه m هستند.

مطابق با الگوریتم تقسیم، اگر $a = q_1 m + r_1$ و $b = q_2 m + r_2$ که در آن $0 \le r_1, r_2 \le m-1$، آنگاه $a \equiv b \bmod m$ اگر و تنها اگر $r_1 = r_2$. به عبارت دیگر $a \equiv b \bmod m$ اگر و تنها اگر a,b در تقسیم بر m دارای باقیمانده های یکسانی باشند، یعنی $a \bmod m = b \bmod m$

مثال. از آنجایی که $101 = 7 \times 14 + 3$، بنابراین $101 \bmod 7 = 3$ و نیز $-4 = 7 \times (-1) + 3$، بنابراین $-4 \bmod 7 = 3$. لذا داریم $101 \equiv -4 \bmod 7$. توجه داشته باشید که $-101 = 7 \times (-15) + 4$، بنابراین $-101 \bmod 7 = 4$.

فرض کنید $P = C = \mathbb{Z}_{26}$ که در آن $\mathbb{Z}_{26}$ باقیمانده اعداد صحیح بر ۲۶ است و $k \in \mathbb{Z}_{26}$ یک مقدار دلخواه باشد. در این صورت تابع رمزگذاری و رمزگشایی در یک سیستم رمز شیفتی (با مقدار شیفت k) به صورت زیر تعریف می شود: (که در آن $x, y \in \mathbb{Z}_{26}$)

$$e_k(x) = (x + k) \bmod 26$$

$$d_k(y) = (y - k) \bmod 26$$

توجه. در حالتی که $k=3$، سیستم رمز شیفتی، همان سیستم سزار است که توسط جولیوس سزار به کار گرفته شد.

تناظر بین حروف و اعداد

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

در واقع ما از رمز شیفتی، برای رمزنگاری متون انگلیسی با استفاده از یک تناظر دوسویی بین حروف انگلیسی و اعداد باقیمانده در پیمانه ۲۶ به صورت جدول فوق استفاده می کنیم. در حالت کلی، می توان از تناظر بین اعداد و حروف استفاده کرد و از رمز شیفتی در پیمانه مطلوب برای رمز کردن یک زبان دلخواه استفاده نمود.

یک مثال از رمز شیفتی

مثال. در سیستم رمز شیفتی، فرض کنید $k=11$ و متن ساده به صورت زیر باشد:

```
wewillmeetatmidnight.
```

در این صورت، در ابتدا، متن فوق را با استفاده از جدول تناظر حروف با اعداد به صورت زیر تبدیل می کنیم:

22	4	22	8	11	11	12	4	4	19
0	19	12	8	3	13	8	6	7	19

حال به هر عدد مقدار ۱۱ را اضافه نموده و در پیمانه ۲۶ محاسبه می کنیم:

7	15	7	19	22	22	23	15	15	4
11	4	23	19	14	24	19	17	18	4

سرانجام، مقادیر حاصل را به حروف متناظر می کنیم تا به متن رمزشده زیر دست یابیم:

HPHTWWXPPELEXTOYTRSE.

دو خاصیت مهم در یک سیستم رمزنگاری

توجه. یک سیستم رمزنگاری اگر بخواهد در عمل مفید باشد، باید در یک سری خواص صادق باشد که در اینجا به دو نمونه از آن اشاره می کنیم:

۱- توابع رمزگذار e_k و رمزگشای d_k می بایست به صورت کارا قابل محاسبه باشند.

۲- شخص مهاجم با فرض مشاهده متن رمزشده y، نباید بتواند به کلید k یا متن ساده x دست یابد.

با یک بیان سربسته، خاصیت دوم، مفهوم "امنیت" را تداعی نموده و به تلاش برای به دست آوردن کلید، به شرط داشتن یک متن رمزشده، "تحلیل رمز" یا "رمزشکنی" می گویند.

تحلیل رمز شیفتی

به سادگی دیده می شود که رمز شیفتی (در پیمانه ۲۶) امن نیست، زیرا می توان با استفاده از حمله جستجوی جامع، به کلید دست یافت. زیرا تنها ۲۶ کلید ممکن وجود دارد که به آسانی می توان تمامی آنها را امتحان کرد تا به متن ساده بامعنی دست یافت.

مثال. متن رمزشده زیر را در نظر بگیرید:

JBCRCLQRWCRVNBJENBWRWN,

با رمزگشایی با کلیدهای $k=1,2,...$ به ترتیب، متون زیر را به دست می آوریم:

```
jbcrclqrwcrvnbjenbwrwn
iabqbkpqvbqumaidmavqvm
hzapajopuaptlzhclzupul
gyzozinotzoskygbkytotk
fxynyhmnsynrjxfajxsnsj
ewxmxglmrxmqiweziwrmri
dvwlwfklqwlphvdyhvqlqh
cuvkvejkpvkogucxgupkpg
btujudijoujnftbwftojof
astitchintimesavesnine
```

حال، با مشاهده متون ساده فوق، به متن ساده و با معنی a stitch in time save nine بر می خوریم. بنابراین کلید برابر با $k=9$ است.

رمز جایگشتی

فرض کنید $P=C=\mathbb{Z}_{26}$. مجموعه کلید K شامل تمامی جایگشت های ممکن از ۲۶ سمبل ممکن $0,1,...,25$ است. برای هر جایگشت $\pi \in K$، تعریف کنید:

$$e_\pi(x)=\pi(x) \quad \text{و} \quad d_\pi(y)=\pi^{-1}(y)$$

جایی که π^{-1} وارون جایگشت π است.

مثال. فرض کنید π یک جایگشت تصادفی به صورت زیر باشد:

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T

n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

که کاراکترهای متن ساده با حروف کوچک انگلیسی و کاراکترهای متن رمزشده با حروف بزرگ انگلیسی نمایش داده شده اند. بنابراین:

$$e_\pi(a)=X, e_\pi(b)=N, ...$$

در این صورت تابع رمزگشا با استفاده از وارون جایگشت π به صورت زیر خواهد بود:

A	B	C	D	E	F	G	H	I	J	K	L	M
d	l	r	y	v	o	h	e	z	x	w	p	t

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	g	f	j	q	n	m	u	s	k	a	c	i

به عبارت دیگر $d_\pi(A)=d, d_\pi(B)=l, \ldots$.

به عنوان یک تمرین، سعی کنید متن رمزشده زیر را که با استفاده از رمز جایگشتی رمز شده است، بشکنید:

MGZVYZLGHCMHJMYXSSFMNHAHYCDLMHA.

توجه دارید که در حمله جستجوی جامع، شکستن یک متن رمز شده با استفاده از رمز جایگشتی نیاز به چک نمودن $26! \approx 4.0 \times 10^{26}$ جایگشت دارد که عدد خیلی بزرگی است. بنابراین، این حمله عملا غیرممکن می باشد (حتی با استفاده از کامپیوترهای پیشرفته). در ادامه خواهیم دید که این سیستم رمزنگاری با استفاده از روش های دیگر حمله به سادگی قابل شکستن است.

رمز آفین

رمز شیفتی، حالت خاصی از رمز جایگشتی است که تنها شامل ۲۶ جایگشت از میان $26!$ جایگشت ممکن روی ۲۶ حرف الفباست. نمونه ای دیگر از رمز جایگشتی، رمز آفین است که با استفاده از تابع رمزگذاری زیر تعریف می شود:

$$e(x) = ax + b \bmod 26$$

جایی که $a,b \in \mathbb{Z}_{26}$. (این نوع توابع، توابع آفین نامیده می شوند). در حالت خاص، اگر $a=1$، آنگاه رمز آفین، همان رمز شیفتی خواهد بود. از طرف دیگر، تابع آفین، باید دوسویی باشد به همین منظور می بایست $\gcd(a,26)=1$. در این حالت تابع رمزگشایی با استفاده از ضابطه زیر به دست می آید:

$$d(y) = a^{-1}(y-b) \bmod 26$$

بنابراین، مجموعه کلید به صورت زیر خواهد بود:

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

مثال از رمز آفین.

فرض کنید $K = (7,3)$. به وضوح می دانیم $7^{-1} \bmod 26 = 15$، زیرا $7 \times 15 = 1 \bmod 26$. بنابراین توابع رمزگذاری و رمزگشایی به صورت زیر می باشند:

$e_k(x) = 7x + 3$ و $d_k(y) = 15(y-3) = 15y - 19$

حال، فرض کنید می خواهیم کلمه hot را رمزگذاری کنیم. ابتدا، حروف h، o و t را به اعداد ۷، ۱۴ و ۱۹ تبدیل می کنیم. بنابراین با استفاده از قاعده رمزگذاری آفین داریم:

$$\begin{aligned}(7 \times 7 + 3) \bmod 26 &= 52 \bmod 26 = 0\\ (7 \times 14 + 3) \bmod 26 &= 101 \bmod 26 = 23\\ (7 \times 19 + 3) \bmod 26 &= 136 \bmod 26 = 6.\end{aligned}$$

که متناظر با رشته AXG است. در اینجا رمزگشایی این رشته را خود تحقیق کنید.

رمزشکنی آفین

برای شکستن رمز آفین در پیمانه m با استفاده از جستجوی جامع نیاز به چک کردن $m \times \phi(m)$ کلید داریم که در آن تابع $\phi(m)$ تابع فی-اویلر نامیده شده و برابر با تعداد اعداد صحیح مثبت کوچکتر از m است که نسبت به آن اول می باشند. به طور مثال، $\phi(26) = 12$ زیرا اعداد مثبت کمتر از ۲۶ که نسبت به آن اول می باشند همان اعداد ۳، ۵، ۷، ۹، ۱۱، ۱۵، ۱۷، ۱۹، ۲۱، ۲۳ و ۲۵ هستند. بنابراین برای $m = 26$ تعداد کلیدهای ممکن برابر با $12 \times 26 = 312$ است. (البته این مقدار برای امن بودن سیستم، خیلی کم است).

در حالت کلی، اثبات می شود که اگر $m = \prod_{i=1}^{n} p_i^{e_i}$ آنگاه $\phi(m)$ به صورت زیر قابل محاسبه است.

$$\phi(m) = \prod_{i=1}^{n} (p_i{}^{e_i} - p_i^{e_i - 1}).$$

به عنوان مثال، برای $m = 60$ داریم:

$$\phi(60) = (4-2) \times (3-1) \times (5-1) = 2 \times 2 \times 4 = 16.$$

رمز ویجینر

هم در رمز شیفتی و هم رمز جایگشتی، پس از انتخاب یک کلید هر حرف الفبا به یک حرف الفبا به صورت یکتا تصویر می شود. به همین خاطر به این سیستم های رمزنگاری، سیستم های تک الفبایی گفته می شود. در اینجا، یک سیستم چند الفبایی معرفی می کنیم که به رمز ویجینر معروف است. (ویجینر در صده شانزدهم زندگی می کرده است).

فرض کنید m یک عدد مثبت باشد. تعریف کنید $P = C = K = (\mathbb{Z}_{26})^m$. برای هر کلید $K = (k_1, k_2, ..., k_m)$، تعریف کنید

$$e_K(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$$

و

$$d_K(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m),$$

جایی که تمامی محاسبات در پیمانه $\mathbb{Z}_{26}$ محاسبه شده است.

مثال از رمز ویجینر

فرض کنید $m = 6$ و کلمه کلید برابر با *CIPHER* باشد. این کلمه متناظر با کلید $K = (2,8,15,7,4,17)$ است. فرض کنید متن ساده به صورت رشته زیر باشد:

thiscryptosystemisnotsecure.

با تبدیل عناصر متن ساده به اعداد در پیمانه ۲۶، می توانیم آنها را به گروه های ۶ تایی تقسیم بندی کرده و سپس کلمه کلید در پیمانه ۲۶ را به آنها به صورت زیر اضافه کنیم:

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15

18	19	4	12	8	18	13	14	19	18	4	2
2	8	15	7	4	17	2	8	15	7	4	17
20	1	19	19	12	9	15	22	8	25	8	19

20	17	4
2	8	15
22	25	19

این الفبا متناظر با متن رمزشده زیر است:

VPXZGIAXIVWPUBTTMJPWIZITWZT.

برای رمزگشایی می توانیم از کلید مشابه استفاده کنیم، منتها به جای جمع کردن، از تفریق کردن در پیمانه ۲۶ استفاده می کنیم.

توجه ۱. تعداد کدکلمات ممکن با طول m در رمز ویجینر برابر با 26^m است، که حتی برای مقادیر کوچک m، جستجوی جامع کلید نیاز به زمان بسیار طولانی خواهد داشت. برای نمونه، به ازای $m=5$ فضای کلید دارای اندازه بیشتر از 1.1×10^7 است. این مقدار مانع از حمله جستجوی جامع به صورت دستی می گردد (البته توسط کامپیوتر این حمله عملی است).

توجه ۲. در رمز ویجینر، یک کاراکتر الفبا می تواند به یکی از m کاراکتر الفبای ممکن تصویر شود (با فرض آنکه کلمه کلید شامل m کاراکتر متمایز باشد). چنین سیستم رمزنگاری، یک سیستم رمزنگاری "چندالفبایی" نامیده می شود. در حالت کلی، رمزشکنی یک سیستم رمزنگاری چند الفبایی نسبت به سیستم رمزنگاری تک الفبایی سخت تر می باشد.

رمز هیل

در اینجا یک سیستم دیگر چندالفبایی به نام رمز هیل را معرفی می کنیم که توسط لستر هیل در سال ۱۹۲۹ مطرح شد. فرض کنید m یک عدد صحیح مثبت باشد و تعریف کنید $P=C=(\mathbb{Z}_{26})^m$. ایده کار، ساخت m کاراکتر الفبای متن رمزشده بر اساس ترکیبات خطی از m کاراکتر متن ساده است. برای همین منظور، فرض کنید کلید $K=(k_{i,j})$ یک ماتریس $m\times m$ باشد. برای $x=(x_1,...,x_m)\in P$، متن رمزشده $y=e_K(x)=(y_1,...,y_m)$ به صورت زیر محاسبه می شود:

$$(y_1, y_2, \ldots, y_m) = (x_1, x_2, \ldots, x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \ldots & k_{1,m} \\ k_{2,1} & k_{2,2} & \ldots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \ldots & k_{m,m} \end{pmatrix}.$$

به عبارت دیگر، با استفاده از نماد ماتریس داریم $y=xK$. از طرف دیگر، به منظور رمزگشایی متن رمزشده y، کافی است از ماتریس وارون K^{-1} استفاده کرده و متن ساده x را با استفاده از رابطه $x=yK^{-1}$ به دست آوریم.

مثال از رمز هیل

مثال. فرض کنید کلید به صورت زیر باشد:

$$\Rightarrow K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}. \qquad K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}.$$

بنابراین، اگر متن ساده به صورت july باشد، آنگاه دو عضو متن ساده به صورت $(9,20)$ (متناظر با ju) و $(11,24)$ (متناظر با ly) است که به صورت زیر رمزگذاری می شود:

$$(9,20)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (99+60, 72+140) = (3,4)$$

$$(11,24)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (121+72, 88+168) = (11,22).$$

بنابراین، متن رمزشده به صورت DELW است. برای رمزگشایی، نیز به صورت مشابه، باب می بایست محاسبات زیر را انجام دهد (تا به متن ساده دست یابد):

$$(11,22)\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (11,24). \qquad (3,4)\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (9,20)$$

شرط وجود تابع رمزگشای هیل

توجه داریم که شرط وجود تابع رمزگشای هیل، وجود ماتریس K^{-1} است که برای این منظور، ماتریس K باید وارون پذیر باشد، اما این مطلب در جبر خطی اثبات شده که شرط وارون پذیری، داشتن دترمینان ناصفر است، یعنی باید $\det(K) \neq 0$. دترمینان ماتریس مربعی A به صورت استقرایی و از بسط دترمینان نسبت به یک سطر یا ستون دلخواه و به صورت زیر به دست می آید:

$$\det A = \sum_{j=1}^{m} (-1)^{i+j} a_{i,j} \det A_{ij},$$

که در آن $A_{i,j}$ ماتریس حاصل از حذف سطر i–ام و ستون j–ام ماتریس A می باشد. به طور مثال، دترمینان ماتریس $A = (a_{i,j})_{n\times n}$، به ازای $n = 2,3$ به صورت زیر به دست می آید:

اگر $n = 2$ آنگاه

$$\det A = a_{1,1}a_{2,2} - a_{1,2}a_{2,1}.$$

و اگر $n = 3$ آنگاه:

$$\begin{aligned} \det A = {} & a_{1,1}a_{2,2}a_{3,3} + a_{1,2}a_{2,3}a_{3,1} + a_{1,3}a_{2,1}a_{3,2} \\ & -(a_{1,1}a_{2,3}a_{3,2} + a_{1,2}a_{2,1}a_{3,3} + a_{1,3}a_{2,2}a_{3,1}). \end{aligned}$$

البته، این روش برای محاسبه دترمینان، کارا نبوده و دارای پیچیدگی بالاست. در واقع، برای محاسبه دترمینان، از روش دیگری استفاده می کنیم که همان اعمال سطری مقدماتی بوده و به طور مفصل در جبرخطی آمده است (برای مطالعه بیشتر به کتاب جبرخطی هافمن مراجعه نمایید).

حال، فرض کنید $\det(K) \neq 0$، برای به دست آوردن K^{-1} کافی است از رابطه $K^{-1} = (\det K)^{-1} K^{*}$ استفاده کنیم که در آن $K^{*} = (k^{*}_{i,j})$ ماتریس الحاقی K نامیده شده و درایه (i,j)–ام آن به صورت

$k^*_{i,j} = (-1)^{i+j} \det K_{j,i}$ به دست می آید، که در آن $K_{j,i}$ ماتریس حاصل از حذف سطر j–ام و ستون i–ام ماتریس K است. به طور مثال اگر $K = (k_{i,j})$ یک ماتریس 2×2 باشد، آنگاه:

$$K^{-1} = (k_{1,1}k_{2,2} - k_{1,2}k_{2,1})^{-1} \begin{pmatrix} k_{2,2} & -k_{1,2} \\ -k_{2,1} & k_{1,1} \end{pmatrix}$$

که البته تمامی محاسبات در پیمانه ۲۶ محاسبه شده است. به عنوان مثال:

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix},$$

رمز جایگشتی

فرض کنید m یک عدد صحیح مثبت باشد. فرض کنید $P = C = (\mathbb{Z}_{26})^m$ و K شامل تمامی جایگشت های $\{1,...,m\}$ باشد. برای هر کلید (جایگشت) π، تعریف کنید:

$$e_\pi(x_1,...,x_m) = (x_{\pi(1)},...,x_{\pi(m)}), \qquad d_\pi(y_1,...,y_m) = (y_{\pi^{-1}(1)},...,y_{\pi^{-1}(m)}),$$

جایی که π^{-1} جایگشت وارون π است.

مثال. فرض کنید $m = 6$ و کلید، جایگشت زیر باشد:

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

.

در این صورت، جایشگت وارون π^{-1} به صورت زیر خواهد بود:

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

.

حال، فرض کنید متن ساده به صورت زیر باشد:

`shesellsseashellsbytheseashore.`

در ابتدا، متن ساده فوق را به گروهی از ۶ حروف به صورت زیر تبدیل می کنیم:

shesel | lsseas | hellsb | ythese | ashore

حال، هر ۶ حرف را با استفاده از جایگشت π، مرتب می کنیم، داریم:

EESLSH | SALSES | LSHBLE | HSYEET | HRAEOS

سرانجام، متن رمزشده به صورت زیر خواهد بود:

EESLSHSALSESLSHBLEHSYEETHRAEOS.

به صورت مشابه، با به کارگیری جایگشت وارون π^{-1} می توان از متن رمزشده به متن ساده متناظر دست یافت.

رمز جایشگتی حالت خاصی از رمز هیل

نکته. رمز جایگشتی، حالت خاصی از رمز هیل است. کافی است، به ازای جایگشت مفروض π روی $\{1,...,m\}$، ماتریس جایگشتی متناظر K_π را به صورت زیر تعریف کنیم:

$$k_{i,j} = \begin{cases} 1 & \text{if } i = \pi(j) \\ 0 & \text{otherwise.} \end{cases}$$

(ماتریس جایگشتی، ماتریسی است که در هر سطر و ستون آن دقیقاً یک ۱ وجود داشته باشد و بقیه عناصر صفر باشند). به سادگی، می توان بررسی کرد که رمزگذاری با استفاده از رمز هیل با ماتریس K_π، معادل همان رمز جایگشتی است. به عنوان نمونه، ماتریس جایگشتی متناظر (به همراه وارون آن) در رمز جایگشتی مثال قبل به صورت زیر است:

$$K_\pi = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad K_\pi^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

رمز دنباله ای

در سیستم های رمزنگاری که تاکنون مشاهده کردیم، عناصر متن ساده به صورت متوالی با استفاده از کلید ثابت K رمزگذاری می شوند. یعنی، رشته متن رمزشده به صورت زیر به دست می آید:

$$\mathbf{y} = y_1 y_2 \cdots = e_K(x_1)e_K(x_2)\cdots .$$

سیستم های رمزنگاری به این صورت را اغلب، رمزهای بلوکی می نامیم. روش دیگر، استفاده از رمزهای دنباله ای (stream cipher) است. ایده اصلی این رمزها، تولید دنباله ای از کلید به صورت $z = z_1 z_2 \ldots$ و استفاده از آن برای رمزنگاری رشته متن ساده $x = x_1 x_2 \ldots$ به صورت زیر است:

$$\mathbf{y} = y_1 y_2 \cdots = e_{z_1}(x_1)e_{z_2}(x_2)\cdots .$$

ساده ترین نمونه از سیستم رمز دنباله ای، آن سیستم رمزی است که رشته کلید از روی یک کلید ساخته شده که مستقل از رشته متن ساده بوده و با یک الگوریتم مشخص تولید می شود. این گونه از سیستم رمزنگاری، سیستم رمز همزمان (synchronous) نامیده شده و به صورت زیر تعریف می شود.

رمز دنباله ای همزمان

یک رشته دنباله ای همزمان، یک چندتایی به صورت (P, C, K, L, E, D) به همراه یک تابع g است، به طوری که شرایط زیر برقرار باشد:

۱. P یک مجموعه متناهی از متن های ساده ممکن است.

۲. C یک مجموعه متناهی از متن های رمزشده ممکن است.

۳. K، فضای کلید، یک مجموعه متناهی از کلیدهای ممکن است.

۴. L یک مجموعه متناهی است که الفبای رشته کلید نامیده می شود.

۵. g مولد رشته کلید نامیده می شود. g کلید K را به عنوان ورودی گرفته و یک رشته نامتناهی $z_1z_2...$ تولید می کند که رشته کلید نامیده می شوند، جایی که $z_i \in L$، برای تمامی $i \geq 1$.

۶. برای هر $z \in L$، یک قاعده رمزنگاری $e_z \in E$ و تابع رمزگشای $d_z \in D$ وجود دارد؛ $e_z : P \to C$ و $d_z : C \to P$ توابعی هستند به طوری که $d_z(e_z(x)) = x$ برای تمامی $x \in P$

رمز ویجینر، حالت خاص رمز دنباله ای

رمز ویجینر، حالت خاصی از رمز دنباله ای همزمان است. برای مشاهده این مطلب، فرض کنید m طول کلمه کلید رمز ویجینر باشد. تعریف کنید $K = (\mathbb{Z}_{26})^m$، $P = C = L = \mathbb{Z}_{26}$، $e_z(x) = (x + z) \bmod 26$ **و** $d_z(y) = (y - z) \bmod 26$. **در آخر، رشته کلید** $z = z_1z_2...$ **را به صورت زیر تعریف کنید:**

$$z_i = \begin{cases} k_i & \text{if } 1 \leq i \leq m \\ z_{i-m} & \text{if } i \geq m+1, \end{cases}$$

جایی که $K = (k_1,...,k_m)$. این کلید، رشته کلید زیر را از تولید می کند:

$$k_1k_2 \cdots k_mk_1k_2 \cdots k_mk_1k_2 \cdots$$

ملاحظه. می توان رمز بلوکی را حالت خاصی از رمز دنباله ای تعریف کرد، کافی است برای هر $i \geq 1$، رشته کلید را $z_i = K$ (یعنی مقدار ثابت کلید در رمز بلوکی) در نظر بگیریم.

یک نمونه رمز دنباله ای همزمان

رمز دنباله ای متناوب. یک رمز دنباله ای، یک رمز دنباله ای متناوب با دوره تناوب d نامیده می شود اگر $z_{i+d} = z_i$، برای تمامی $i \geq 1$. رمز ویجینر با طول کلید m (آنچنان که قبلا تشریح شد) یک رمز دنباله ای متناوب با دوره تناوب m است.

رمز دنباله ای دودویی. رمزهای دنباله ای، اغلب بر حسب الفبای دوتایی توصیف می شوند، یعنی $P = C = L = \mathbb{Z}_2$. در این حالت، عملگرهای رمزگذاری و رمزگشایی در پیمانه ۲ محاسبه می شوند.

$$e_z(x) = (x+z) \bmod 2, \quad d_z(y) = (y+z) \bmod 2.$$

در اینجا می خواهیم یک روش دیگر برای تولید رشته کلید (همزمان) مطرح کنیم. فرض کنید یک رشته m–تایی $(k_1,...,k_m)$ داده شده است و $z_i = k_i$ برای $1 \le i \le m$. در این صورت می توانیم رشته کلید را با استفاده از یک رابطه بازگشتی خطی به صورت زیر بسازیم:

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

برای تمامی $i \ge 1$، جایی که $c_0,...,c_{m-1} \in \mathbb{Z}_2$ مقادیر ثابتی هستند.

ملاحظه. اولاً این رابطه، از درجه m است، زیرا هر جمله به m جمله قبلی وابسته است. نیز، این رابطه خطی است، زیرا z_{i+m} یک ترکیب خطی برحسب جملات ماقبل است. همچنین، بدون کاستن از کلیت، می توانیم فرض کنیم $c_0 = 1$، چرا که در غیراین صورت، رابطه بازگشتی از درجه حداکثر $m-1$ خواهد بود.

در اینجا، کلید K شامل $2m$ مقدار $k_1,...,k_m,c_0,...,c_{m-1}$ است. اگر

$$(k_1, \ldots, k_m) = (0, \ldots, 0),$$

آنگاه، رشته کلید تماماً شامل عناصر ۰ خواهد بود، که مطلوب نیست؛ زیرا در این حالت، متن ساده با متن رمزشده یکسان خواهد بود. در غیراین صورت، $(k_1,...,k_m)$ دارای یک دوره تناوب خواهد بود، که در بهترین حالت دوره تناوب آن برابر با $2^m - 1$ خواهد بود که مطلوب است. دراین حالت، یک کلید "کوتاه"، منجر به یک رشته کلید متناوب با دوره تناوب "طولانی" $2^m - 1$ خواهد شد.

رجیسترتغییرمکان یا LFSR

مثال. فرض کنید $m = 4$ و رشته کلید با استفاده از رابطه بازگشتی زیر تعریف شده باشد:

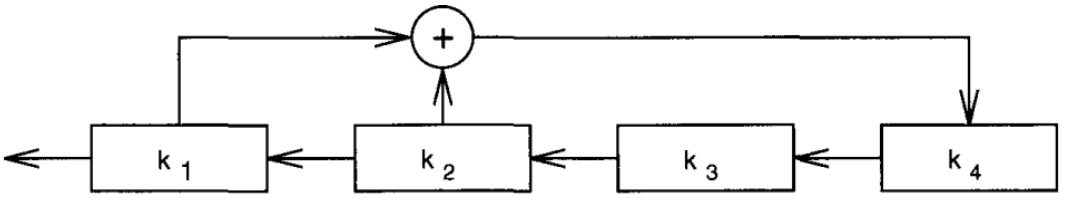
$$z_{i+4} = (z_i + z_{i+1}) \bmod 2,$$

اگر رشته کلید با هر بردار غیر از $(0,0,0,0)$ مقدار اولیه شده باشد، آنگاه رشته کلید با دوره تناوب ۱۵ را خواهیم داشت. برای نمونه، با شروع از بردار $(1,0,0,0)$ ، رشته کلید زیر تولید خواهد شد:

$$1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\cdots.$$

هر بردار ناصفر دیگر نیز اگر به عنوان بردار اولیه قرار بگیرد، یک جایگشت دوری از رشته کلید زیر را نتیجه خواهد داد.

رجیستر تغییر مکان. یکی از جنبه های مثبت روش فوق، نمایش رشته کلید با استفاده از رجیسترهای تغییرمکان روبه عقب خطی (linear feedback shift register یا LFSR) است، که یک رجیستر تغییر مکان با m مرحله است.



رمز دنباله ای غیرهمزمان

در واقع، بردار $(k_1,...,k_m)$ به عنوان یک بردار اولیه برای رجیستر تغییر مکان به کار می رود.

در واقع، بردار $(k_1,...,k_m)$ به عنوان یک بردار اولیه برای رجیستر تغییر مکان به کار می رود.

در هر واحد زمانی، عملگرهای زیر به طور متوالی صورت می پذیرد:

۱. k_1 به بیت رشته کلید بعدی انتقال می یابد.
۲. $k_2,...,k_m$ هر یک به خانه بعدی (سمت چپ) شیفت داده می شود.
۳. مقدار جدید k_m به صورت زیر محاسبه می شود:

$$\sum_{j=0}^{m-1} c_j k_{j+1}$$

(که یک رابطه خطی است).

در هر لحظه از زمان، رجیستر تغییر مکان شامل m عضو رشته کلید متوالی، مانند $z_i,...,z_{i+m-1}$ است. در این حالت، در واحد زمان بعدی، رجیستر تغییر مکان شامل $z_{i+1},...,z_{i+m}$ می باشد.

رمز دنباله ای غیرهمزمان: یک رمز دنباله ای است که هر عضو رشته کلید z_i تنها به متن ساده قبلی $(x_1,...,x_{i-1})$ یا عناصر متن رمزشده قبلی $(y_1,...,y_{i-1})$ به همراه کلید K بستگی دارد.

یک نمونه ساده از رمز دنباله ای غیرهمزمان، رمز خودکلید (autokey cipher) است.

رمز خود کلید

فرض کنید $P=C=K=L=\mathbb{Z}_{26}$. فرض کنید $z_1=K$ و تعریف کنید $z_i=x_{i-1}$ برای تمامی $i\geq 2$.

برای $0\leq z\leq 25$ ، تعریف کنید: $(x,y\in\mathbb{Z}_{26})$

$$d_z(y)=(y-z) \bmod 26 \qquad e_z(x)=(x+z) \bmod 26$$

مثال. فرض کنید $K=8$ و متن ساده به صورت زیر باشد:

rendezvous.

ما در ابتدا این متن ساده را به دنباله ای اعداد صحیح زیر تبدیل می کنیم.

17 4 13 3 4 25 21 14 20 18

رشته کلید به صورت زیر می باشد:

8 17 4 13 3 4 25 21 14 20

حال، با جمع عناصر متناظر در پیمانه ۲۶، دنباله رمزشده زیر را خواهیم داشت:

25 21 17 16 7 3 20 9 8 12

که با تبدیل آن به حروف الفبا، متن رمزشده زیر را خواهیم داشت:

ZVRQHDUJIM.

حال، نگاهی به کدگشایی می اندازیم. در ابتدا، ما متن رمزشده را به رشته اعداد زیر تبدیل می کنیم:

$$25 \quad 21 \quad 17 \quad 16 \quad 7 \quad 3 \quad 20 \quad 9 \quad 8 \quad 12$$

حال، متن ساده شده را به صورت زیر محاسبه می کنیم:

$$x_1 = d_8(25) = (25 - 8) \bmod 26 = 17.$$

به همین صورت، در ادامه:

$$x_2 = d_{17}(21) = (21 - 17) \bmod 26 = 4,$$

و همین طور ادامه می دهیم. هر لحظه ای که ما کاراکتر متن ساده شده بعدی را به دست می آوریم، از آن به عنوان عضو رشته کلید بعدی استفاده می کنیم.

رمزشکنی یا تحلیل رمز

در اینجا، برخی روش های مورد استفاده در شکستن یک رمز را مورد تجزیه و تحلیل قرار می دهیم.

از این فرض استفاده می کنیم که اسکار، به عنوان حمله کننده، می داند چه سیستم رمزی مورد استفاده قرار گرفته است (اصل کرشهف). در این حالت، اگر اسکار نداند چه سیستم رمزی استفاده شده است، با سختی بیشتری می تواند سیستم رمز را بشکند. متداول ترین مدل های حمله به شرح زیر می باشد:

۱- **حمله تنها متن رمزشده:** فرد مهاجم تنها به رشته ای از متن رمزشده y دسترسی دارد.

۲- **حمله تنها متن ساده:** فرد مهاجم به رشته ای از متن های ساده x به همراه متن رمزشده متناظر با آنها y دسترسی دارد.

۳- **حمله متن ساده معلوم:** فرد مهاجم، دسترسی موقتی به دستگاه رمزنگاری دارد. بنابراین، او می تواند یک متن ساده x را انتخاب نموده و متن رمزشده متناظر با آن y را بسازد.

۴- **حمله متن رمزشده معلوم:** فرد مهاجم، دسترسی موقتی به دستگاه رمزشکنی دارد. بنابراین، او می تواند یک رشته متن رمزشده y را انتخاب نموده و متن ساده متناظر با آن x را بسازد.

حمله آماری

هر مرحله، هدف تعیین کلید مورد استفاده است؛ این مطلب به مهاجم، اجازه می دهد تا رشته متن رمزشده "هدف" را کدگشایی کند و علاوه بر آن قادر به شکستن هر رشته متن رمزشده ای است که با آن کلید رمزشده است.

در ابتدا، ضعیف ترین نوع حمله، یعنی حمله تنها متن رمزشده، را در نظر بگیرید. در اینجا، همچنین، فرض می کنیم که رشته متن ساده حروف الفبای انگلیسی (بدون نقطه یا فاصله یا کاما) باشد.

letter	probability	letter	probability
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001

۱- حمله آماری:

بکر (Beker) و پیپر (Piper) جدول زیر را برای تخمین فرکانس (احتمال متوسط حروف در کلمات با معنی انگلیسی مانند مجلات، دیکشنری ها) مورد استفاده قرار دادند. در این جدول، این افراد، ۲۶ حرف انگلیسی را در ۵ رده به صورت زیر طبقه بندی کردند:

فرکانس حروف

۱. حرف E که دارای بیشترین احتمال، حدود ۰/۱۲۰ است.

۲. حروف T، A، I،O ، N، S، H، R که هر کدام دارای احتمالی مابین ۰/۰۶ و ۰/۰۹ هستند.

۳. D، L هر کدام دارای احتمالی قریب به ۰/۰۴ هستند.

۴. C، U، M، W، F، G، Y، P، B هرکدام دارای احتمالی مابین ۰/۰۱۵ و ۰/۰۲۸ هستند.

۵. V، K، J، X، Q، Z هرکدام دارای احتمالی کمتر از ۰/۰۱ هستند.

همچنین، مفید است که دنباله حروف متوالی مشمتمل بر ۲ یا ۳ حرف را در نظر بگیریم، که به ترتیب دو حرفی ها و سه حرفی ها نامیده می شوند. ۳۰ تا از متداول ترین دو حرفی ها به صورت زیر می باشند:

TH, HE, IN, ER, AN, RE, ED, ON, ES, ST,
EN, AT, TO, NT, HA, ND, OU, EA, NG, AS,
OR, TI, IS, ET, IT, AR, TE, SE, HI, OF.

همچنین، ۱۲ تا از بیشترین سه تایی ها به صورت زیر می باشند:

THE, ING, AND, HER, ERE, ENT,
THA, NTH, WAS, ETH, FOR, DTH.

شکستن رمز آفین

نمونه ای از حمله آماری را می توان در شکستن رمز آفین (مثال زیر) مشاهده کرد.

مثال. فرض کنید متن رمزشده زیر با استفاده از رمز آفین به دست آمده است:

```
FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDK
APRKDLYEVLRHHRH
```

تحلیل فرکانس این متن رمزشده را می توان در جدول زیر مشاهده کرد:

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

در این متن رمزشده، تنها ۵۷ کاراکتر وجود دارد، اما همین تعداد نیز کافی است تا بتوانیم رمز آفین را رمزگشایی کنیم. کاراکترهای متن رمزشده با بیشترین فرکانس برابر با R (۸ تا)، D (۷ تا)، E، H، K (۵ تا) و F، S، V (۴ تا) می باشد. در اولین مرحله، فرض می کنیم R به e رمزگذاری شده و D به t رمزگذاری می شود؛ چون e و t (به ترتیب) متداول ترین حروف می باشد. در نمایش به وسیله اعداد، داریم $e_K(4)=17$ و $e_K(19)=3$. حال از آنجایی که $e_k(x)=ax+b$، با جایگذاری می توانیم a,b را به صورت زیر بیابیم:

$$4a+b=17$$
$$19a+b=3.$$

این دستگاه، دارای جواب یکتای $a=6, b=19$ در $\mathbb{Z}_{26}$ است. اما این جواب غیرقابل قبول است، زیرا $\gcd(a,26)=2>1$. بنابراین، فرض اولیه ما غلط است. حدس دیگر، این است که R به e و E به t باید رمزنگاری شود. با این فرض به دست می آوریم $a=13$ که مجدداً غیرقابل قبول است. در حدس دیگر، R را به e و H را به t رمزگشایی می کنیم که در این حالت $a=3, b=5$ که در فرض اولیه یک کلید قابل قبول به نظر می رسد. حال، اگر با این کلید رمزگشایی کنیم، تابع رمزگشای $d_K(y)=9y-19$ و متناظر با آن رشته زیر را خواهیم داشت که یک متن بامعنی است؛

algorithmsarequitegeneraldefinitionsofarit
hmeticprocesses

رمزگشایی رمز جانشینی

در اینجا، با رمزشکنی رمز جانشینی که نسبت به رمز آفین پیچیده تر است، از طریق مثال زیر آشنا می شویم.

مثال. متن رمزشده زیر که با استفاده از رمز جانشینی رمزگذاری شده را در نظر بگیرید:

```
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

جدول زیر فرکانس حروف را در این متن رمزشده نشان می دهد:

letter	frequency	letter	frequency
A	0	N	9
B	1	O	0
C	15	P	1
D	13	Q	4
E	7	R	10
F	11	S	3
G	1	T	2
H	4	U	5
I	5	V	5
J	11	W	8
K	1	X	6
L	0	Y	10
M	16	Z	20

از آنجایی که Z بیشترین بار رخ داده است، می توان حدس زد $d_K(Z) = e$. سایر کاراکترهای متن رمزشده که حداقل ۱۰ بار رخ داده باشند به صورت C,D,F,J,M,R,Y هستند. می توان تصور کرد این حروف، رمزگذاری شده حروف t,a,o,l,n,s,h,r باشند، اما تعداد وقوع این حروف آن قدر زیاد نیست که بتوان دقیقاً حروف متناظر با آنها را تعیین کرد.

در این مرحله، نگاهی به دوحرفی ها می اندازیم که به شکل z–یا z- باشند (با این فرض که Z به e رمزگشایی شود). متداول ترین دوحرفی ها در متن رمزشده به صورت DZ و ZD (هر کدام چهار مرتبه)، NZ و ZU (هر کدام سه مرتبه) و RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD, ZJ (هر کدام دو مرتبه) هستند. از آنجایی که ZW چهار مرتبه

رخ داده و WZ اصلا رخ نداده است، و W نسبت به سایر کاراکترها کم رخ داده است، می توان حدس زد $d_K(W)=d$. از آنجایی که DZ چهار مرتبه رخ داده و ZD دو مرتبه، می توان تصور کرد $d_K(D)\in\{r,s,t\}$ ، اما این که کدام یک از این سه احتمال، درست است، واضح نیست.

اگر با این فرض ادامه دهیم که $d_K(Z)=e, d_K(W)=d$، آنگاه می توان به متن رمزشده بازگشت و توجه نمود که ZRW تقریباً در ابتدای متن رمزشده رخ داده و RW مجدداً پس از آن رخ داده است. از آنجایی که R در متن رمزشده زیاد رخ داده و nd یک دو حرفی متداول است، ممکن است تصور کنیم $d_K(R)=n$ (با بیشترین احتمال).

در این حالت، با جایگذاری در متن رمز گشایی شده، متن زیر را داریم:

```
------end---------e----ned---e------------
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

--------e----e---------n--d---en----e----e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

-e---n------n------ed---e---e--ne-nd-e-e--
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

-ed-----n-----------e----ed-------d---e--n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

در مرحله بعدی، ممکن است تصور کنیم $d_K(N)=h$، زیرا NZ یک دو حرفی متداول است و ZN چنین نیست. اگر این فرض صحیح باشد، آنگاه قسمتی از متن ساده که به صورت ne-ndhe رمزگشایی شده، پیشنهاد $d_K(C)=a$ را به ما می دهد. اگر این فرضیات را کنار هم بگذاریم، متن رمزگشایی شده به صورت زیر خواهد بود:

```
------end-----a---e-a--nedh--e------a-----
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h-------ea---e-a---a---nhad-a-en--a-e-h--e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he-a-n------n------ed---e---e--neandhe-e--
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

-ed-a---nh---ha---a-e----ed-----a-d--he--n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

حال، M را به عنوان دومین حرف با بیشترین تکرار در متن رمزشده در نظر بگیرید. سه حرفی RNM که به -nh رمزگشایی شده، این پیشنهاد را به ما می دهد که -h ابتدای یک کلمه بوده است، بنابراین M احتمالاً نمایش یک حرف صدادار بوده است. اما، از آنجایی که قبلاً برخی حروف متن رمزشده به a و e رمزگشایی شده، پس امیدواریم $d_K(M)=i,o$. از طرف دیگر، دو حرفی ai نسبت به ao بیشترین رخ می دهد، پس با توجه به CM، ابتدا فرض می کنیم $d_K(M)=i$. در این صورت، متن رمزشده به صورت زیر کدگشایی خواهد شد:

```
-----iend-----a-i-e-a-inedhi-e------a---i-
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h-----i-ea-i-e-a---a-i-nhad-a-en--a-e-hi-e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he-a-n-----in-i----ed---e---e-ineandhe-e--
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

-ed-a--inhi--hai--a-e-i--ed-----a-d--he--n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

در ادامه، به دنبال حرفی می گردیم که به o رمزگشایی شده باشد. از آنجایی که o یک حرف پرتکرار در متن ساده است، انتظار داریم تا کاراکتر متن رمز متناظر با آن یکی از حروف D,F,J,Y باشد. به نظر می رسد که Y محتمل ترین باشد، چرا که در غیر این صورت رشته های طولانی از حروف صدادار به صورت aoi از CFM یا CJM خواهیم داشت. بنابراین، بیایید فرض کنیم $d_K(Y)=o$. سه حرف باقیمانده با بیشترین احتمال، به صورت D,F,J می باشد. دو بار تکرار سه تایی NMD پیشنهاد می دهد $d_K(D)=s$ ، که این فرض، سه تایی his را می دهد (این فرض نیز با فرض اولیه $d_K(D)\in\{r,s,t\}$ سازگار است). ۵ حرفی HNCMF می تواند به chair رمزگشایی شود که بنابراین $d_K(F)=r$ و $d_K(H)=c$. پس داریم $d_K(J)=t$ (با استفاده از فرآیند حذف). بنابراین، متن رمزگشایی شده به صورت زیر خواهد بود:

```
o-r-riend-ro--arise-a-inedhise--t---ass-it
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

hs-r-riseasi-e-a-orationhadta-en--ace-hi-e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he-asnt-oo-in-i-o-redso-e-ore-ineandhesett
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

-ed-ac-inhischair-aceti-ted--to-ardsthes-n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

در ادامه، به سادگی می توان سایر حروف متن ساده را نیز حدس زد که متن رمزگشایی شده به صورت زیر خواهد بود:

> Our friend from Paris examined his empty glass with surprise, as if evaporation had taken place while he wasn't looking. I poured some more wine and he settled back in his chair, face tilted up towards the sun.[2]

رمزشکنی رمز ویجینر

در اینجا، به دنبال روش برای شکستن رمز ویجینر هستیم. در مرحله اول، باید طول کلید را بیابیم که با m نمایش داده می شود. روش های مختلفی برای این کار وجود دارد که معروف ترین آن روش کاسیسکی (۱۸۶۳) است و روش دیگر استفاده از اندیس تطابق (۱۹۲۰) است. مبنای روش کاسیسکی این مطلب است که دو قطعه از متن ساده به متن ساده به طور یکسان رمزگذاری می شوند، هرگاه محل وقوع آنها در متن ساده به اندازه δ مکان با یکدیگر فاصله داشته باشد، جایی که $\delta \equiv 0 (\mathrm{mod}\, m)$. برعکس، اگر مشاهده کنیم که دو قطعه از متن رمزشده یکسان و با طول حداقل ۳ باشند، آنگاه با احتمال بالایی، آنها متناظر با قطعات یکسانی از متن ساده هستند.

تست کاسیسکی به صورت زیر است. در ابتدا به دنبال قطعات یکسان از متن رمزشده می گردیم و فاصله بین مکان های شروع آنها را به دست می آوریم؛ فرض کنید این فواصل به صورت $\delta_1, \delta_2, \ldots$ باشد، در این صورت حدس می زنیم که m می بایست تمامی این δ_i ها را عاد کند و لذا m می بایست بزرگترین مقسوم علیه مشترک آنها را عاد کند. روش اندیس تطابق نیز به شرح زیر است.

تعریف. فرض کنید $x = x_1 x_2 \ldots x_n$ یک رشته شامل n حرف الفبا باشد. اندیس تطابق x که با $I_c(x)$ نمایش داده می شود، احتمال این است که دو عضو تصادفی x یکسان باشند.

فرض کنید فرکانس حروف A,B,C,...,Z در x به ترتیب به صورت $f_0,...,f_{25}$ باشد. می توانیم دو عضو متفاوت x را با $\binom{n}{2}$ روش انتخاب کنیم. برای هر $0 \le i \le 25$، به تعداد $\binom{f_i}{2}$ روش وجود دارد که هر دو عضو انتخاب شده برابر با i باشند. بنابراین، فرمول زیر را داریم:

$$I_c(\mathbf{x}) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}.$$

اندیس تطابق

فرض کنید x رشته ای از حروف انگلیسی باشد و $p_0,...,p_{25}$ به ترتیب احتمال وقوع حروف A,B,..,Z باشد. در این صورت:

$$I_c(\mathbf{x}) \approx \sum_{i=0}^{25} {p_i}^2 = 0.065,$$

زیرا احتمال این که دو حرف انتخاب شده A باشد، برابر با p_0^2 ، دو حرف انتخاب شده B باشد، برابر با p_1^2 و ...

به طریق مشابه، این رابطه درحالتی که x یک رشته متن رمزشده حاصل از متن رمز تک الفبایی باشد، برقرار است و مقدار $\sum p_i^2$ ثابت باقی می ماند. حال، فرض کنید رشته متن رمزشده $y = y_1...y_n$ حاصل از رمز ویجینر باشد. رشته y را به m زیررشته $y_1,...,y_m$ تقسیم کنید (با نوشتن متن رمزشده در یک آرایه با n/m سطر و m ستون به صورت سطربه سطر). به عبارت دیگر داریم:

$$\begin{aligned}
\mathbf{y}_1 &= y_1 y_{m+1} y_{2m+1} \cdots, \\
\mathbf{y}_2 &= y_2 y_{m+2} y_{2m+2} \cdots, \\
&\vdots \\
\mathbf{y}_m &= y_m y_{2m} y_{3m} \cdots.
\end{aligned}$$

اگر $y_1,...,y_m$ به این روش ساخته شوند، و m طول واقعی کلید باشد، آنگاه هر مقدار $I_c(y_i)$ باید تقریباً برابر با ۰/۰۶۵ باشد. از طرف دیگر، اگر m طول کلید واقعی نباشد، آنگاه زیررشته y_i به نظر بیشتر تصادفی می رسد، زیرا این رشته ها با رمز شیفتی با کلیدهای متفاوت به دست آمده اند. در حالت تصادفی داریم:

$$I_c \approx 26\left(\frac{1}{26}\right)^2 = \frac{1}{26} = 0.038.$$

مثال برای به دست آوردن طول رمز ویجینر

دو مقدار ۰/۰۶۵ و ۰/۰۳۸ به اندازه کافی از یکدیگر فاصله دارند به گونه ای که در اغلب اوقات خواهیم توانست طول کلید صحیح را با استفاده از این روش تشخیص دهیم (یا این که حدسی را که قبلا با آزمون کاسیسکی زده ایم امتحان کنیم). حال بیایید این دو روش را با استفاده از یک مثال توضیح دهیم.

مثال. متن رمزشده زیر با استفاده از رمز ویجینر را در نظر بگیرید:

```
CHREEVOAHMAERATBIAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK
LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX
VRVPRTULHDNQWTWDTYGBPHXTFALJHASVBFXNGLLCHR
ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT
AMRVLCRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI
PEEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP
WQAIIWXNRMGWOIIFKEE
```

در ابتدا، آزمون کاسیسکی را به کار می بریم. زیر رشته CHR در ۵ جای متن فوق به کار رفته است. (مکان های ۱، ۱۶۶، ۲۳۶، ۲۷۶ و ۲۸۶). فاصله این مکان ها به ترتیب برابر با ۱۶۵، ۲۳۵، ۲۷۶ و ۲۸۵ است. بزرگترین مقسوم علیه این چهار مقدار صحیح برابر با ۵ است که به نظر طول کلید می رسد.

حال، ببینیم با استفاده از اندیس تطابق، آیا همین نتیجه به دست می آید. برای m=1 ، اندیس تطابق برابر با ۰/۰۴۵ است . برای m=2 دو اندیس تطابق برابر با ۰/۰۴۶ و ۰/۰۴۱ هستند. برای m=3، این مقادیر برابر با ۰/۰۴۳، ۰/۰۵۰ و ۰/۰۴۷ می باشند. برای m=4، این مقادیر برابر با ۰/۰۴۲، ۰/۰۳۹، ۰/۰۴۵ و ۰/۰۴۰ هستند. اما برای m=5 این مقادیر برابر با ۰/۰۶۳، ۰/۰۶۸، ۰/۰۶۹، ۰/۰۶۱ و ۰/۰۷۲ می باشند. این مطلب، نشان می دهد m=5 صحیح است.

به دست آوردن کلید

حال فرض کنید توانستیم طول کلید رمز ویجنیر، یعنی مقدار m، را به درستی بیابیم. در ادامه به دنبال یافتن کلید صحیح $K = (k_1, \ldots, k_m)$ هستیم. برای این منظور یک روش ساده و موثر را توضیح می دهیم. فرض کنید

$1 \le i \le m$ و $f_0,...,f_{25}$ به ترتیب معرف فرکانس حروف A,B,...,Z در رشته y_i باشند. نیز، فرض کنید $n' = n/m$ معرف طول رشته y_i باشد. در این صورت، توزیع احتمال ۲۶ حرف در y_i به صورت زیر خواهد بود:

$$\frac{f_0}{n'}, \dots, \frac{f_{25}}{n'}.$$

یادآوری می کنیم که زیررشته y_i با رمزگذاری شیفتی زیرمجموعه ای از متن ساده به اندازه شیفت k_i به دست آمده است. بنابراین، انتظار داریم تا این مقادیر شیفت یافته دارای توزیع احتمالاتی زیر باشند:

$$\frac{f_{k_i}}{n'}, \dots, \frac{f_{25+k_i}}{n'}$$

که نزدیک به توزیع احتمالاتی $p_0,...,p_{25}$ می باشد (اندیس ها در فرمول بالا در پیمانه ۲۶ محاسبه شده اند.)

حال، برای $0 \le g \le 25$ تعریف کنید:

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

اگر $g = k_i$، آنگاه انتظار داریم تا:

$$M_g \approx \sum_{i=0}^{25} {p_i}^2 = 0.065,$$

همچنان که در اندیس تطابق این مطلب در نظر گرفته شد. اگر $g \neq k_i$ آنگاه M_g به طور قابل ملاحظه ای کمتر از ۰/۰۶۵ خواهد بود. با به کارگیری این روش امیدواریم بتوانیم به k_i صحیح دسترسی پیدا کنیم.

مثال. در مثال قبل، طول کلید را برابر با ۵ حدس زدیم. حال می خواهیم مقادیر M_g را برای $1 \le i \le 5$ به دست آوریم و برای هر i به دنبال مقداری از M_g می گردیم که نزدیک به ۰/۰۶۵ باشد. این مقدار g شیفت های $k_1,...,k_5$ را تعیین خواهد کرد. این مقادیر در جدول زیر لیست شده اند:

i	value of $M_g(\mathbf{y}_i)$								
1	.035	.031	.036	.037	.035	.039	.028	.028	.048
	.061	.039	.032	.040	.038	.038	.045	.036	.030
	.042	.043	.036	.033	.049	.043	.042	.036	
2	.069	.044	.032	.035	.044	.034	.036	.033	.029
	.031	.042	.045	.040	.045	.046	.042	.037	.032
	.034	.037	.032	.034	.043	.032	.026	.047	
3	.048	.029	.042	.043	.044	.034	.038	.035	.032
	.049	.035	.031	.035	.066	.035	.038	.036	.045
	.027	.035	.034	.034	.036	.035	.046	.040	
4	.045	.032	.033	.038	.060	.034	.034	.034	.050
	.033	.033	.043	.040	.033	.029	.036	.040	.044
	.037	.050	.034	.034	.039	.044	.038	.035	
5	.034	.031	.035	.044	.047	.037	.043	.038	.042
	.037	.033	.032	.036	.037	.036	.045	.032	.029
	.044	.072	.037	.027	.031	.048	.036	.037	

با توجه به این مقادیر در جدول فوق، داریم $K = (9,0,13,4,19)$، بنابراین کلید برابر با JANET است. با رمزگشایی متن رمزشده با این کلید، متن رمزگشایی شده به صورت زیر خواهد بود:

The almond tree was in tentative blossom. The days were longer, often ending with magnificent evenings of corrugated pink skies. The hunting season was over, with hounds and guns put away for six months. The vineyards were busy again as the well-organized farmers treated their vines and the more lackadaisical neighbors hurried to do the pruning they should have done in November.[4]

شکستن رمز هیل

شکستن رمز هیل با حمله تنها متن رمزشده کار سختی است، اما می توان آن را به آسانی با استفاده از حمله متن ساده معلوم شکست. بیایید فرض کنیم که دشمن به مقدار m به کار برده شده دسترسی دارد. فرض کنید او حداقل m زوج متن ساده زیر به همراه متن رمزشده متناظر با آنها را می داند.

$$y_j = (y_{1,j}, y_{2,j}, \ldots, y_{m,j}), \qquad x_j = (x_{1,j}, x_{2,j}, \ldots, x_{m,j})$$

که در آن $1 \le j \le m$ و نیز $y_j = e_K(x_j)$. حال، اگر دو ماتریس $m \times m$ به صورت $X = (x_{i,j})$ و $Y = (y_{i,j})$ را تشکیل دهیم، آنگاه معادله $Y = XK$ را خواهیم داشت، جایی که ماتریس K کلید است که باید تعیین شود. به شرط آن که X وارون پذیر باشد، اسکار می تواند کلید را با استفاده از رابطه $K = X^{-1}Y$ به دست آورد. (اگر X

وارون پذیر نباشد، باید مجموعه دیگری از زوج های شامل m متن ساده به همراه متن رمزشده متناظر با آنها را امتحان کنیم.)

مثال. فرض کنید متن ساده Friday با استفاده از رمز هیل با m=2 به متن PQCFKU رمزشده است. در اینجا داریم $e_K(5,17)=(15,16)$ و $e_K(0,24)=(10,20)$. حال با توجه به زوج های متن ساده-متن رمزشده، معادله ماتریسی زیر را داریم.

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K.$$

حال، به سادگی داریم:

$$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix},$$

$$K = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

رمزشکنی رمز دنباله ای

نکته. اگر دشمن، مقدار m را نداند، چه باید انجام داد؟ در این حالت، با فرض این که m خیلی بزرگ نباشد، می توان مقادیر m=2,3,... را به سادگی امتحان کرد تا به کلید دست یافت. اگر مقدار m حدس زده درست نباشد، آنگاه ماتریس $m \times m$ یافت شده با الگوریتم فوق با زوج متن ساده-متن رمزشده داده شده مطابقت نخواهد کرد.

رمزشکنی رمز دنباله ای بر پایه LFSR

یادآوری می کنیم که رمز دنباله ای، مجموع متن ساده و رشته کلید در پیمانه ۲ است، یعنی:

$$y_i = (x_i + z_i) \bmod 2$$

رشته کلید نیز با استفاده از یک mتایی اولیه $(z_1,\ldots,z_m)=(k_1,\ldots,k_m)$ و با استفاده از رابطه بازگشتی زیر به دست می آید:

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

جایی که $c_0,...,c_{m-1} \in \mathbb{Z}_2$. از آنجایی که تمامی عملیات در این سیستم رمزنگاری خطی می باشد، ممکن است گمان کنیم که این سیستم رمزنگاری همانند رمز هیل نسبت به حمله متن ساده معلوم آسیب پذیر است. فرض کنید اسکار رشته ای از متن ساده به صورت $x_1...x_n$ و متن رمزشده متناظر با آنها $y_1...y_n$ دسترسی دارد. در این صورت، او می تواند بیت های رشته کلید را به صورت $z_i = (x_i + y_i) \bmod 2$ به دست آورد. حال بیایید فرض کنیم اسکار مقدار m را می داند، بنابراین، کافی است تا او مقادیر $c_0,...,c_{m-1}$ را به دست آورد (تا بتواند کل رشته کلید را تولید کند).

برای این منظور، توجه داریم برای $i \geq 1$:

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

که یک دستگاه معادله خطی با m مجهول است. حال اگر $n \geq 2m$ آنگاه m معادله خطی با m مجهول خواهیم داشت که قابل حل است. این دستگاه m معادله را می توان به فرم ماتریسی زیر نوشت:

$$(z_{m+1}, z_{m+2}, \ldots, z_{2m}) = (c_0, c_1, \ldots, c_{m-1}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}.$$

اگر ماتریس ضرایب در پیمانه ۲ وارون پذیر باشد، آنگاه جواب زیر را به دست می آوریم:

$$(c_0, c_1, \ldots, c_{m-1}) = (z_{m+1}, z_{m+2}, \ldots, z_{2m}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}^{-1}.$$

در واقع، این ماتریس وارون پذیر است، هرگاه m درجه معادله بازگشتی مورد استفاده برای تولید رشته کلید باشد.

مثال برای شکستن رمز دنباله ای

مثال. فرض کنید اسکار به رشته متن رمزشده زیر دسترسی دارد:

$$101101011110010$$

که متناظر با رشته متن ساده زیر است:

$$011001111111000.$$

در این صورت، او می تواند بیت های رشته کلید را به صورت زیر محاسبه کند:

$$110100100001010.$$

نیز، فرض کنید اسکار می داند که رشته کلید با استفاده از یک LFSR با ۵ خانه تولید شده است. در این صورت ، او باید معادله ماتریسی زیر را حل کند که از ۱۰ بیت رشته کلید به دست آمده است:

$$(0,1,0,0,0) = (c_0, c_1, c_2, c_3, c_4) \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

به سادگی، می توان دید:

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix},$$

بنابراین، داریم:

$$\begin{aligned} (c_0, c_1, c_2, c_3, c_4) &= (0,1,0,0,0) \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \\ &= (1,0,0,1,0). \end{aligned}$$

بنابراین، رابطه بازگشتی مورد استفاده برای رشته کلید به صورت زیر است:

$$z_{i+5} = (z_i + z_{i+3}) \bmod 2.$$

اهداف دشمن: تعیین آن چیزی که برای دشمن، معنی "حمله" به سیستم را می دهد.

- حمله کامل: یافتن کلید
- حمله جزیی: رمزگشایی قسمتی از متن رمزشده (یا یافتن اطلاعاتی جزیی درباره متن ساده)
- توانایی تشخیص: تشخیص بین متن رمزشده معتبر و رشته های تصادفی

سطح امنیت: تشخیص منابع محاسباتی قابل دسترس برای دشمن

- امنیت بدون شرط: منابع محاسباتی نامحدود
- امنیت محاسباتی: اندازه گیری میزان تلاش محاسباتی لازم، با استفاده از روش های شناخته شده، برای شکست سیستم
- امنیت اثبات پذیر: نشان دادن این که سختی شکستن یک سیستم لزوماً به اندازه سختی حل یک مساله سخت معروف (عموماً در نظریه اعداد) است.

سیستم امن در عمل. در عمل یک سیستم معمولا امن نامیده می شود اگر هزینه شکستن آن از مقدار اطلاعات به دست آمده بیشتر باشد یا زمان لازم برای شکستن آن سیستم از زمان عمر اطلاعات بیشتر باشد. نیز، مدت زمان حمله نباید از مدت زمان حمله جستجوی جامع کلید بیشتر باشد.

تقسیم بندی متن های رمزشده (ciphers):

- برمبنای نوع عملگر:

۱- جانشینی: هر عضو متن ساده (بیت، حرف یا گروهی از بیت ها یا حروف) به یک عضو دیگر نگاشته شوند.

۲-جایگشت: اعضای متن ساده با یکدیگر جابجا شوند.

- برمبنای تعداد کلید به کاررفته:

۱- یک کلید برای فرستنده و گیرنده (رمز متقارن)

۲- دو کلید متفاوت (رمز کلید عمومی)

- بر مبنای تعداد متن ساده به کار رفته.

۱- رمز بلوکی: یک بلوک در ورودی در یک لحظه برای تولید یک بلوک در خروجی.

۲- رمز رشته ای: ورودی به طور پیوسته برای تولید یک عضو خروجی در هر واحد زمانی پردازش می شود.

توجه داریم سیستم رمز خودکلید، نوع تغییریافته رمز ویجینر است که در آن کلید خود متن ساده است که به اندازه ثابتی شیفت یافته است. نیز، رمز ویجنینر را در حالتی که بتوانیم طول کلید را بیابیم، قابل شکست خواهد بود. در رمز خودکلید، کلید دارای طول یکسان با متن ساده است. از آنجایی که این مطلب باعث می شود تا خواص آماری متن ساده با خواص آماری متن رمزشده مرتبط باشد، این سیستم نیز قابل شکست به نظر می رسد.

در حالت ایده آل، کلید می بایست دارای طول یکسانی با متن ساده، اما غیرمرتبط با آن، باشد. این مطلب در سیستم صفحه کلید رمز یک رویه (One-time pad) لحاظ شده است.

صفحه کلید رمز یک رویه:

۱. $n \geq 1, P = C = K = (\mathbb{Z}_2)^n$

۲. رمزگذاری: $e_K(x) = (x_1 + k_1, \ldots, x_n + k_n) \bmod 2$

۳. رمزگشایی: $d_K(y) = (y_1 + k_1, \ldots, y_n + k_n) \bmod 2$

مزایا: این سیستم رمز، دارای امنیت کامل است.

معایب:

- کلید (که باید به صورت مخفی مبادله شود) دارای طولی (بزرگ) به اندازه متن ساده است.
- هر کلید تنها می تواند یکبار به کار رود.
- آسیب پذیری در برابر حمله متن ساده معلوم
- مشکلات نگهداری و مدیریت کلید: در تجارت استفاده نمی شود، اما در مکان های نظامی و موقعیت های سیاسی ممکن است استفاده شود.
- بیشتر برای خط ارتباطی مسکو-واشنگتن استفاده شده است.
- بیشتر برای آژانس های روسیه برای تبادل با کشورهای خارجی استفاده شده است.
- صفحه کلید رمز یک رویه توسط ورنام در سال ۱۹۱۸ مطرح شد و سال ها تصور می شد این رمز غیرقابل شکست است تا این که شانون در سال ۱۹۴۹ آن را اثبات کرد. ایده پشت سر آن، استقلال کلید بود که باعث می شد متن رمزشده بتواند به هرچیزی رمزگشایی شود!
- مثال. یک متن ساده را دوبار با استفاده از رمز ورنام رمزگذاری کرده ایم:

```
ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
pxlmvmsydoftyrvzwc tnlebnecvgdupahfzzlmnyih
mr mustard with the candlestick in the hall

ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
pftgpmaydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
miss scarlet with the knife in the library
```

- در امنیت بدون شرط، فرض کردیم که حمله کننده دارای منابع محاسباتی نامحدود باشد.
- برای مطالعه امنیت بدون شرط، نیاز داریم تا برخی مفاهیم نظریه احتمالات را مرور کنیم.

نمادگذاری ها:

- X و Y متغیرهای تصادفی گسسته هستند.

- $\Pr(x) = \Pr(X = x)$: احتمال این که X مقدار x را اختیار کند.

- $\Pr(y) = \Pr(Y = y)$: احتمال این که Y مقدار y را اختیار کند.

- $\Pr(x, y)$: احتمال توام- احتمال این که X مقدار x و Y مقدار y را اختیار کنند.

- $\Pr(x \mid y)$: احتمال شرطی- احتمال این که X مقدار x را اختیار کند، به شرط آن که Y مقدار y را اختیار کرده باشد.

- X و Y مستقل هستند، اگر $\Pr(x, y) = \Pr(x)\Pr(y)$ برای تمامی x, y

- همواره داریم: $\Pr(x, y) = \Pr(x \mid y)\Pr(y) = \Pr(y \mid x)\Pr(x)$

قضیه (قاعده بیز) اگر $\Pr(y) > 0$ آنگاه $\Pr(x \mid y) = \dfrac{\Pr(y \mid x)\Pr(x)}{\Pr(y)}$

نتیجه: X و Y مستقل هستند اگر و تنها اگر $\Pr(x \mid y) = \Pr(x)$ برای تمامی x, y ها

مثال. یک زوج تاس را به صورت تصادفی پرتاب کنید. فرض کنید X متغیر تصادفی مجموع مقادیر این دو تاس و Y دو مقدار D و N را می گیرد، به ترتیب اگر دو مقدار مشاهده شده یکسان یا غیریکسان باشند. توزیع احتمالاتی X و Y در جداول زیر نمایش داده شده اند:

x	2	3	4	5	6	7	8	9	10	11	12
$\text{Prob}(\mathbf{X} = x)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

y	D	N
$\text{Prob}(\mathbf{Y} = y)$	6/36	30/36

دو احتمال شرطی به صورت زیر محاسبه می شوند:

$$\text{Prob}(D|4)(= \text{Prob}(\mathbf{Y} = D|\mathbf{X} = 4)) = 1/3 \qquad \text{Prob}(4|D)(= \text{Prob}(\mathbf{X} = 4|\mathbf{Y} = D)) = 1/6$$

در این حالت، احتمال توام به صورت زیر خواهد بود:

$$\text{Prob}(4, D) = 1/36 = \text{Prob}(D|4)\,\text{Prob}(4) = \text{Prob}(4|D)\,\text{Prob}(D)$$

نمادگذاری ها: فرض کنید یک سیستم رمزنگاری (P, C, K, E, D) داریم.

- $\Pr(X=x)$: احتمال (پیشین) این که متن ساده برابر با x باشد.
- $\Pr(K=k)$: احتمال این که کلید k انتخاب شده باشد.

فرض: K و X متغیرهای تصادفی مستقل هستند.

- $\Pr(Y=y)$: احتمال این که متن رمزشده برابر با y باشد.
- $C(k)=\{e_k(x): x\in P\}$ – تمام متون رمزشده به دست آمده با کلید k

داریم:

$$\text{Prob}(\mathbf{y}=y)=\sum_{\{K|y\in C(K)\}}\text{Prob}(\mathbf{K}=K)\,\text{Prob}(\mathbf{x}=d_K(y))$$

نیز:

$$\text{Prob}(\mathbf{y}=y|\mathbf{x}=x)=\sum_{\{K|x=d_K(y)\}}\text{Prob}(\mathbf{K}=K)$$

حال، با به کارگیری قاعده بیز می توانیم احتمال متن ساده به شرط داشتن یک متن رمزشده را به صورت زیر به دست آوریم:

$$\text{Prob}(\mathbf{x}=x|\mathbf{y}=y)=\frac{\text{Prob}(\mathbf{x}=x)\sum_{\{K|x=d_K(y)\}}\text{Prob}(\mathbf{K}=K)}{\sum_{\{K|y\in C(K)\}}\text{Prob}(\mathbf{K}=K)\,\text{Prob}(\mathbf{x}=d_K(y))}$$

مثال. فرض کنید یک سیستم رمزنگاری با $P=\{a,b\}$ ، $C=\{1,2,3,4\}$ و $K=\{k_1,k_2,k_3\}$ با توزیع احتمال زیر را داریم:

x	a	b
$\text{Prob}(\mathbf{x}=x)$	1/4	3/4

K	K_1	K_2	K_3
$\text{Prob}(\mathbf{K}=K)$	1/2	1/4	1/4

نیز، فرض کنید تابع رمزنگاری به صورت زیر باشد:

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

در این صورت، می توانیم احتمال های زیر را محاسبه کنیم:

y	1	2	3	4
$\text{Prob}(\mathbf{y} = y)$	1/8	7/16	1/4	3/16

$\text{Prob}(\mathbf{x} = x \mid \mathbf{y} = y)$	1	2	3	4
a	1	1/7	1/4	0
b	0	6/7	3/4	1

$\text{Prob}(\mathbf{y} = y \mid \mathbf{x} = x)$	a	b
1	1/2	0
2	1/4	1/2
3	1/4	1/4
4	0	1/4

یک سیستم رمزنگاری دارای امنیت کامل است هرگاه $\Pr(X = x \mid Y = y) = \Pr(X = x)$ برای تمامی x, y ها، یعنی احتمال (پسین) این که متن ساده برابر با x باشد، به شرط آنکه متن رمزشده برابر با y باشد، همیشه با احتمال (پیشین) این که متن ساده برابر با x باشد، برابر باشد. در غیراین صورت، y هیچ اطلاعاتی درباره x به ما نمی دهد.

توجه دارید که این مطلب با توجه به قاعده بیز با $\Pr(Y = y \mid X = x) = \Pr(Y = y)$ معادل است.

قضیه: فرض کنید از یک رمز شیفتی استفاده کرده ایم که هر کاراکتر با استفاده از یک کلید با احتمال یکنواخت (احتمال $\frac{1}{26}$) رمزگذاری شده است. در این صورت، برای هر توزیع متن ساده، رمز شیفتی دارای امنیت کامل خواهد بود.

اثبات: یادآوری می کنیم که $P = C = K = \mathbb{Z}_{26}$ و $e_K(x) = x + K \bmod 26$. برای هر متن رمزشده y داریم:

$$\begin{aligned}
\text{Prob}(\mathbf{y} = y) &= \sum_{K \in \mathbb{Z}_{26}} \text{Prob}(\mathbf{K} = K)\, \text{Prob}(\mathbf{x} = d_K(y)) \\
&= \sum_{K \in \mathbb{Z}_{26}} \frac{1}{26}\, \text{Prob}(\mathbf{x} = y - K) \\
&= \frac{1}{26} \sum_{K \in \mathbb{Z}_{26}} \text{Prob}(\mathbf{x} = y - K) \\
&= \frac{1}{26} \sum_{y \in \mathbb{Z}_{26}} \text{Prob}(\mathbf{x} = y) \\
&= \frac{1}{26}.
\end{aligned}$$

بنابراین، رمز شیفتی با توزیع کلید یکنواخت برای هر حرف دارای امنیت کامل است. نیز داریم:

$$\text{Prob}(\mathbf{y} = y | \mathbf{x} = x) = \text{Prob}(\mathbf{K} = y - x \mod 26) = \frac{1}{26}$$

برای هر y ، فرض کنید $\Pr(Y = y) > 0$ (در غیراین صورت می توانیم y را از متن رمزشده C حذف کنیم). برای یک مقدار ثابت $x \in P$ ، اگر یک سیستم رمزنگاری دارای امنیت کامل باشد، داریم:

$$\Pr(Y = y \mid X = x) = \Pr(Y = y) > 0$$

بنابراین، $k \in K$ وجود دارد به طوری که $e_k(x) = y$. در نتیجه، $|K| \geq |C|$. نیز، رمزگذاری یک به یک است، پس $|C| \geq |P|$.

قضیه (شانون) اگر $|P| = |C| = |K|$ آنگاه سیستم رمزنگاری دارای امنیت کامل است اگر و تنها اگر

۱. تمامی کلیدهای به کار رفته دارای احتمال یکسان باشند.

۲. برای هر $x \in P$ و $y \in C$ ، یک کلید یکتای $k \in K$ وجود داشته باشد به طوری که $e_k(x) = y$

اثبات: در ابتدا، فرض کنید سیستم رمزنگاری دارای امنیت کامل باشد.

۲.در بالا نشان دادیم که برای هر $x \in P$ و $y \in C$ حداقل یک $k \in K$ وجود دارد به طوری که $e_k(x) = y$. اما $|K| = |C|$ که نتیجه می دهد دقیقاً یک کلید به این صورت وجود دارد.

۱. $y \in C$ را ثابت در نظر بگیرید و قرار دهید $P = \{x_1, ..., x_n\}$. می توانیم مجموعه کلیدها را با $\{k_1, ..., k_n\}$ نشان دهیم به طوری که $e_{k_i}(x_i) = y$ برای $1 \le i \le |P|$. در این صورت ، با استفاده از خاصیت امنیت کامل داریم $\Pr(K = k_i) = \Pr(Y = y \mid X = x_i)$ ، برای تمامی i ها. معنی این مطلب، آن است که تمامی کلیدهای به کار رفته دارای توزیع یکسان $\Pr(y)$ می باشند.

عکس این مطلب در قضیه قبل بیان شده است.

نتیجه. سیستم کوله پشتی یکبار یک سیستم دارای امنیت کامل است.

DES یک سیستم رمز متداول است که معمولا بلوک های ۶۴ بیتی را با استفاده از یک کلید ۵۶ بیتی به بلوک های ۶۴ بیتی رمزگذاری می کند.

تاریخچه.

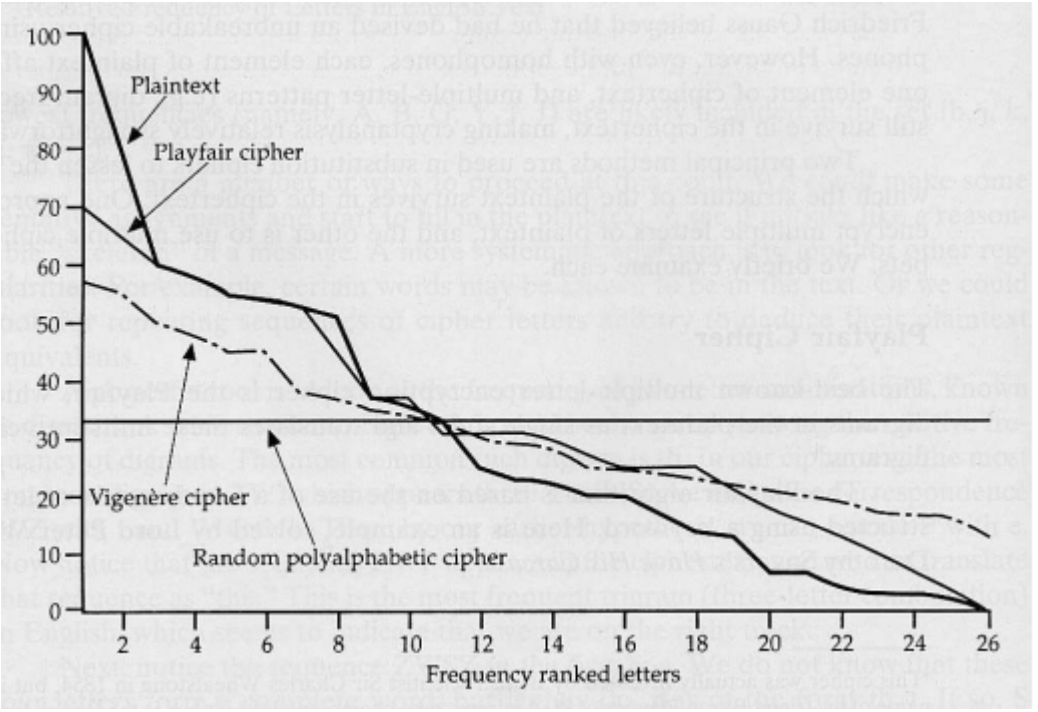
- ۱۹۶۰: فیستل (زیرمجموعه IBM) رمز بلوکی فیستل-لوسیفر را طراحی کرد که روی بلوک های ۶۴ و با استفاده از یک کلید ۱۲۸ بیتی عمل می کردند.
- ۱۹۷۳: NBS یک پروپزال برای استاندارد رمز بین المللی پیشنهاد داد.
- یک رمز لوسیفر بهبود یافته (توسط IBM و NSA) دریافت شد (مورد نظر NSA). این رمز همان DES بود.
- برخی انتقادها:

۱. کلید به کار رفته برای یک حمله جستجوی جامع خیلی کوچک بود.

۲. محک طراحی s-box ها به طور علنی فاش نشد.

- ۱۹۹۴: NIST رمز DES را برای کاربردهایی بیشتر از حفاظت از داده های طبقه بندی شده پیشنهاد داد.
- ۱۹۹۹: NIST تنها سیستم رمز سه تایی DES (دو یا سه کلید DES) پیشنهاد داد.
- روش شکستن رمزی که تاکنون معرفی کردیم، بر مبنای خواص آماری بود. سیستم تک الفبایی به آسانی شکسته می شود، زیرا خواص آماری به خوبی (در سطح حروف) کار می کند. سیستم های چندالفبایی نیز می توانند شکسته شوند، زیرا ما هنوز می توانیم تحلیل

آماری داشته باشیم. در شکل زیر می توانیم ببینیم که چگونه فرکانس حروف از متن ساده به متن رمزشده برای سیستم های رمز متعدد تغییر می کند. بجز برای یک رمز چندالفبایی تصادفی، در هر سیستم رمز دیگری، اطلاعاتی از متن ساده در متن رمزشده باقی می ماند که می توان از آنها برای شکستن سیستم استفاده کرد. در حالت ایده آل، هیچ اطلاعاتی درباره متن ساده یا کلید را نمی توان در متن رمزشده مشاهده کرد. این مطلب، در سیستم رمز کوله پشتی یکبار مشاهده شد، اما در آنجا طول کلید قابل قبول نبود. برای دست یافتن به یک روش کارا که دارای کلید کوچکتری باشد، می توانیم از رمزهای بلوکی استفاده کنیم که همچنان که خواهیم دید با تکرار یک سری مراحل، رمزهای دنباله ای را شبیه سازی می کنند. ایده اصلی این کار به رمزهای فیستل گونه برمی گردد.



-

در عمل، می توانیم نگاشتی را به کار بریم که بلوک های n بیتی را به بلوک های n بیتی تصویر می کند. اما اندازه کلید با 2^n متناسب است که آن را غیرعملی خواهد کرد. به طور نمونه برای غلبه بر خواص آماری برای طول بلوک ۶۴ بیت، طول کلید می بایست تقریباً 10^{19} باشد. بنابراین، باید به دنبال یک روش دیگر باشیم که به تاثیر یکسان دست یابیم. برای این منظور، ایده فیستل را می توان به کار برد.

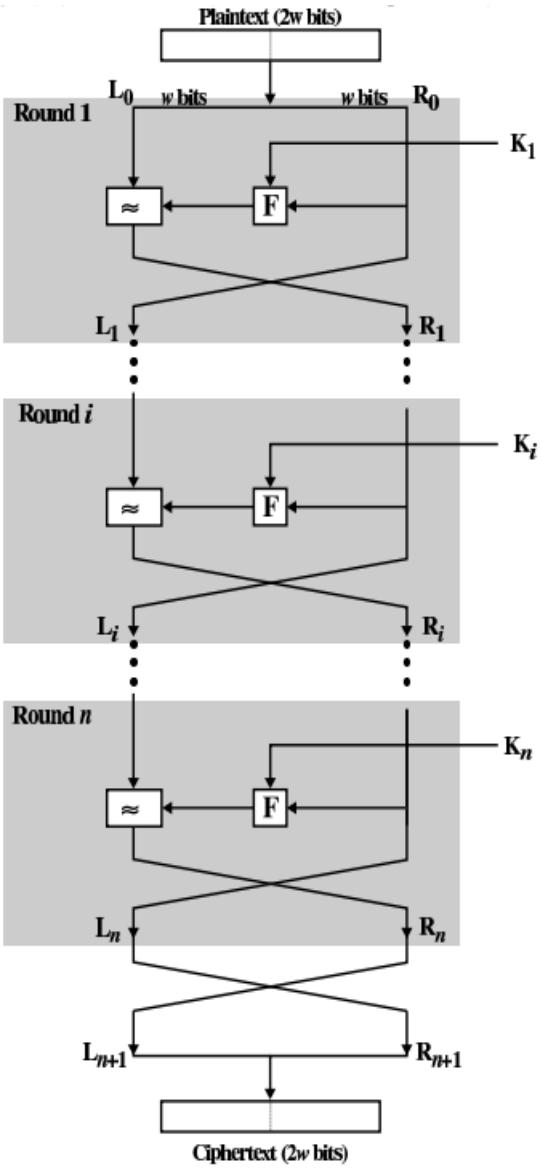
قبل از آن دو ایده اساسی شانون برای برهم زدن خواص آماری را مطرح می کنیم: یعنی انتشار (diffusion) و اغتشاش (confusion).

انتشار یعنی خواص آماری متن ساده باید در خواص آماری محدوده وسیعی از متن رمزشده پراکنده شود. برای نمونه، هر بیت از متن ساده باید روی بیت های زیادی از متن رمز شده تاثیر بگذارد یا به طور هم ارز هر بیت از متن رمزشده متاثر از تعداد زیادی بیت متن ساده باشد. بنابراین، انتشار سعی می کند تا رابطه آماری مابین متن ساده و متن رمزشده را تا حد ممکن پیچیده کند. انتشار توسط جایگشت های تکراری قابل حصول است.

اغتشاش سعی می کند تا رابطه آماری بین متن رمزشده و کلید را تا حد ممکن پیچیده سازد. اغتشاش توسط توابع جانشینی پیچیده قابل حصول است.

ایده اصلی رمز فیستلی در شکل سمت راست نمایش داده شده است. این شکل، حالت خاصی از شبکه جایگشتی مطرح شده توسط شانون است که دارای مراحل زیر است:

- یک تابع جانشینی در سمت چپ داده وجود دارد؛ تابع مرحله F بر نیمه راست اثر کرده و نتیجه با نیمه چپ xor می شود؛ هر مرحله F به مقداری زیرکلید k_i وابسته است.
- یک جایگشت؛ دو نیمه چپ را با یکدیگر جابجا می کند.



پارامترهای مهم در رمز فیستل به شرح زیر می باشند:

اندازه بلوک ها: هر چه بیشتر باشد بهتر است؛ ۶۴ بیت خوب است؛ AES ۱۲۸ بیت را به کار می برد.

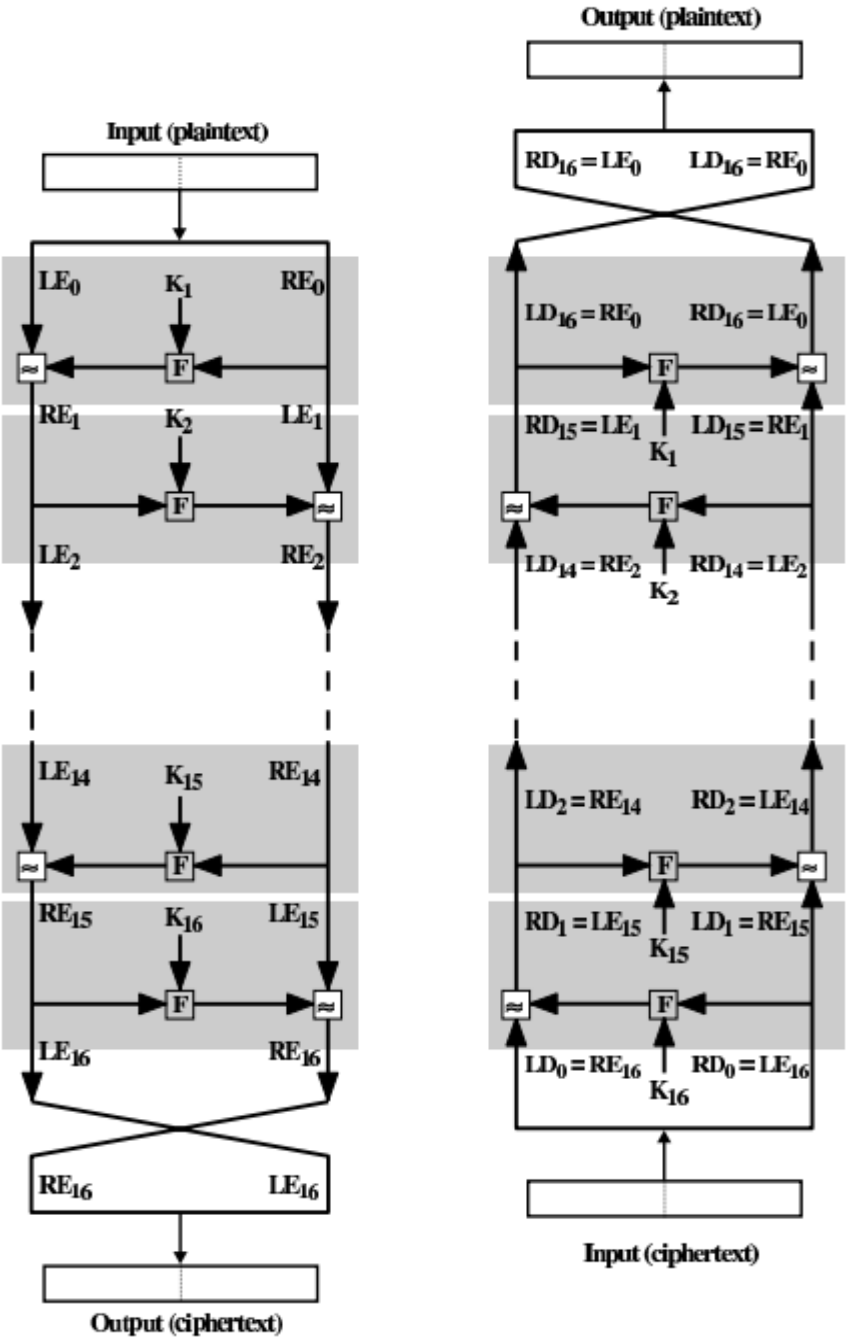
اندازه کلید: بزرگ تر بودن آن امنیت را افزایش می دهد، اما سرعت را کاهش می دهد؛ ۶۴ بیت به اندازه کافی خوب نیست؛ ۱۲۸ بیت اندازه متداولی است.

تعداد مراحل: هر چه بیشتر باشد در مقابل حملات پیشرفته تر مقاوم تر است؛ اندازه متداول ۱۶ است.

الگوریتم تولید زیرکلیدها-بهتر است پیچیده باشد.

تابع مرحله: بهتر است پیچیده باشد.

الگوریتم های رمزنگاری و رمزگشایی لزوماً یکسان هستند، با این تفاوت که زیرکلیدهای به کار برده شده در الگوریتم رمزگشایی به صورت برعکس به کار می روند. برای مشاهده بهتر، شکل سمت راست را مشاهده کنید.



در اینجا، نشان می دهیم که رمزگشایی همان گونه که انتظار می رود، کار می کند. با نمادگذاری موجود در شکل، برای هر i داریم:

$$\begin{aligned} LE_i &= RE_{i-1} & \qquad LD_i &= RD_{i-1} \\ RE_i &= LE_{i-1} \oplus F(RE_{i-1}, K_i) & \qquad RD_i &= LD_{i-1} \oplus F(RD_{i-1}, K_{17-i}) \end{aligned}$$

با استقرا روی i نشان می دهیم که:

$$LD_i = RE_{16-i} \qquad RD_i = LE_{16-i}$$

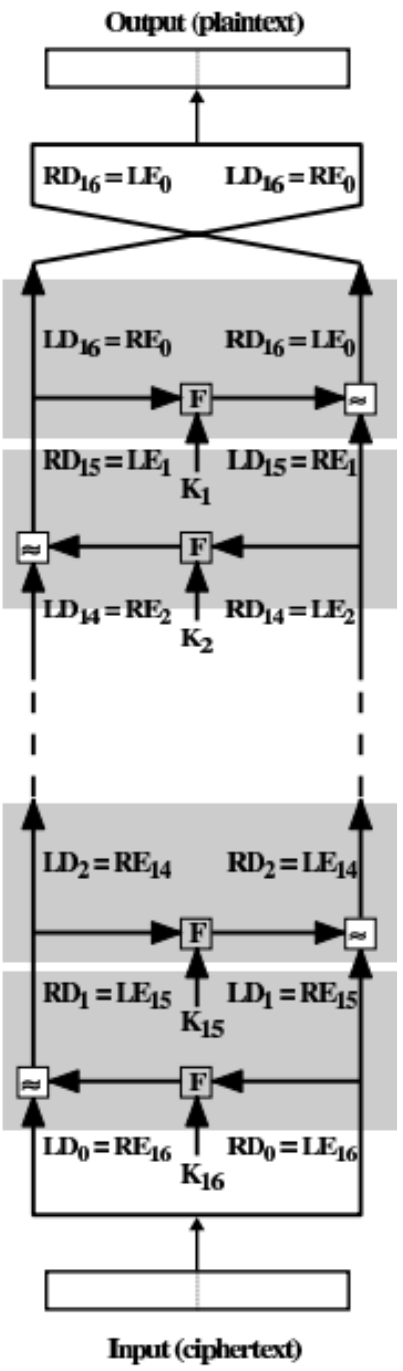
به خصوص، برای i=16، فرآیند رمزگشایی به ما متن ساده را می دهد. تساوی ها برای حالت i=0 برقرار است. فرض می کنیم این روابط برای i برقرار باشد و آن را برای i +1 نشان می دهیم. از این مطلب بهره می جوییم که عمل $\oplus$ شرکت پذیر بوده و دارای عضو همانی ۰ است و این که برای هر x داریم $x \oplus x = 0$. داریم:

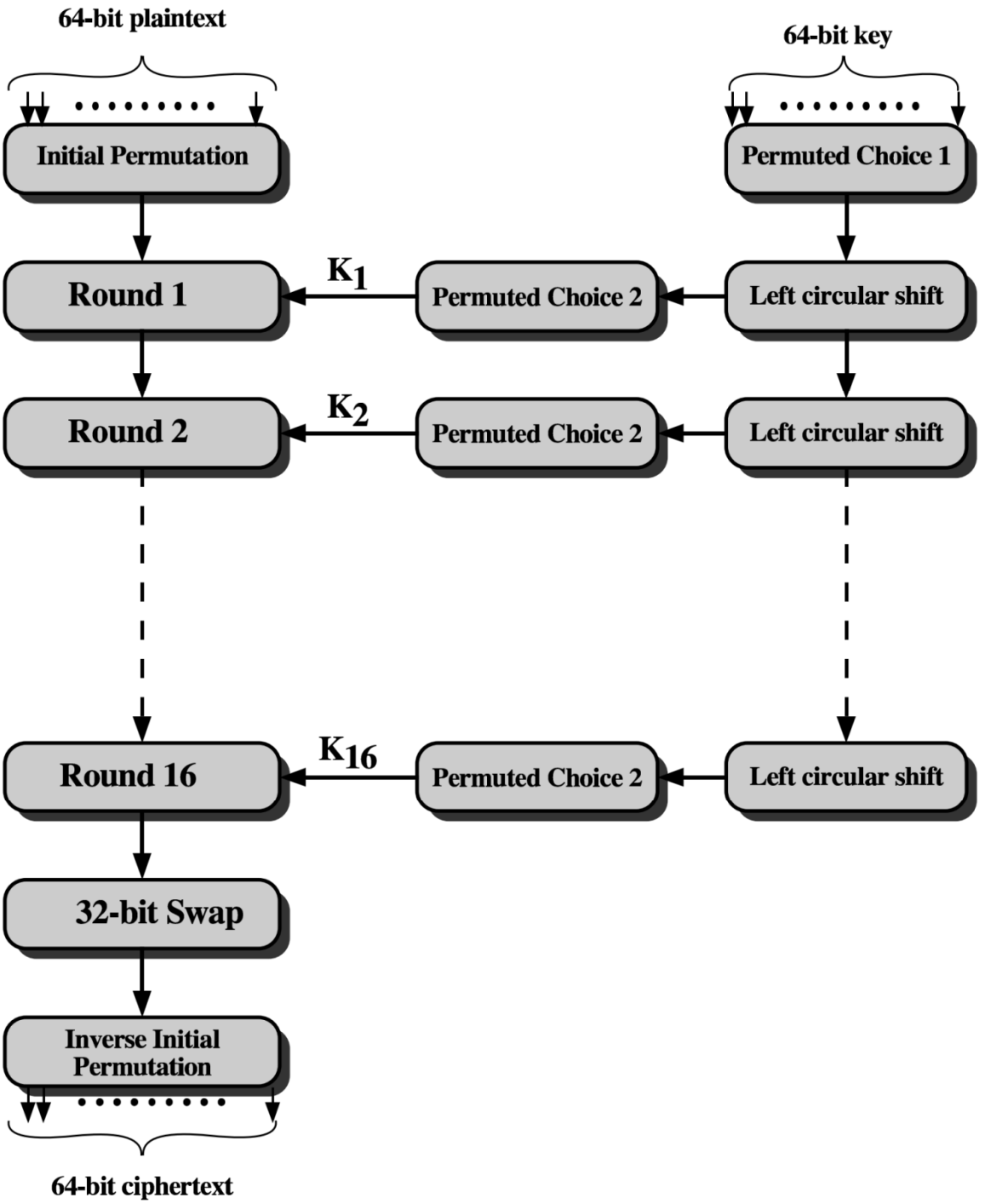
$$LD_{i+1} = RD_i = LE_{16-i} = RE_{16-(i+1)}$$

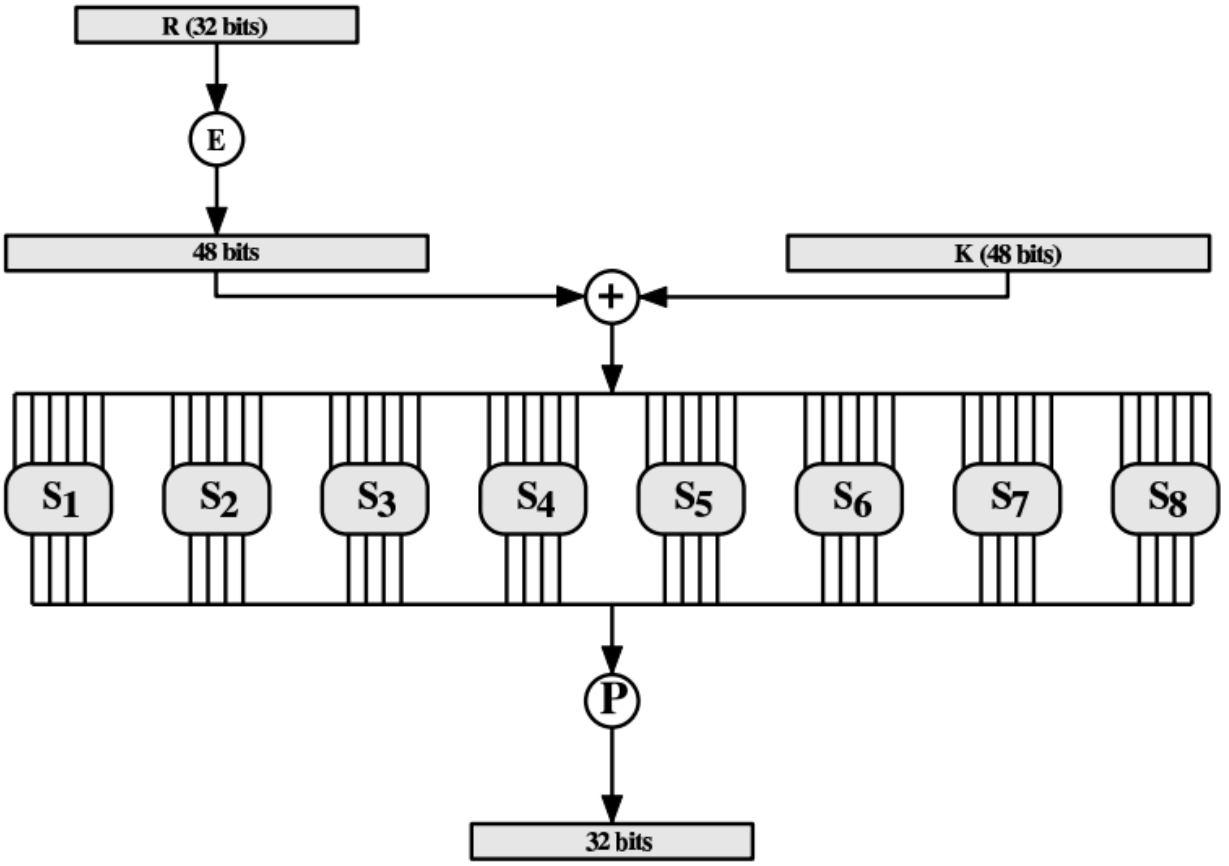
و

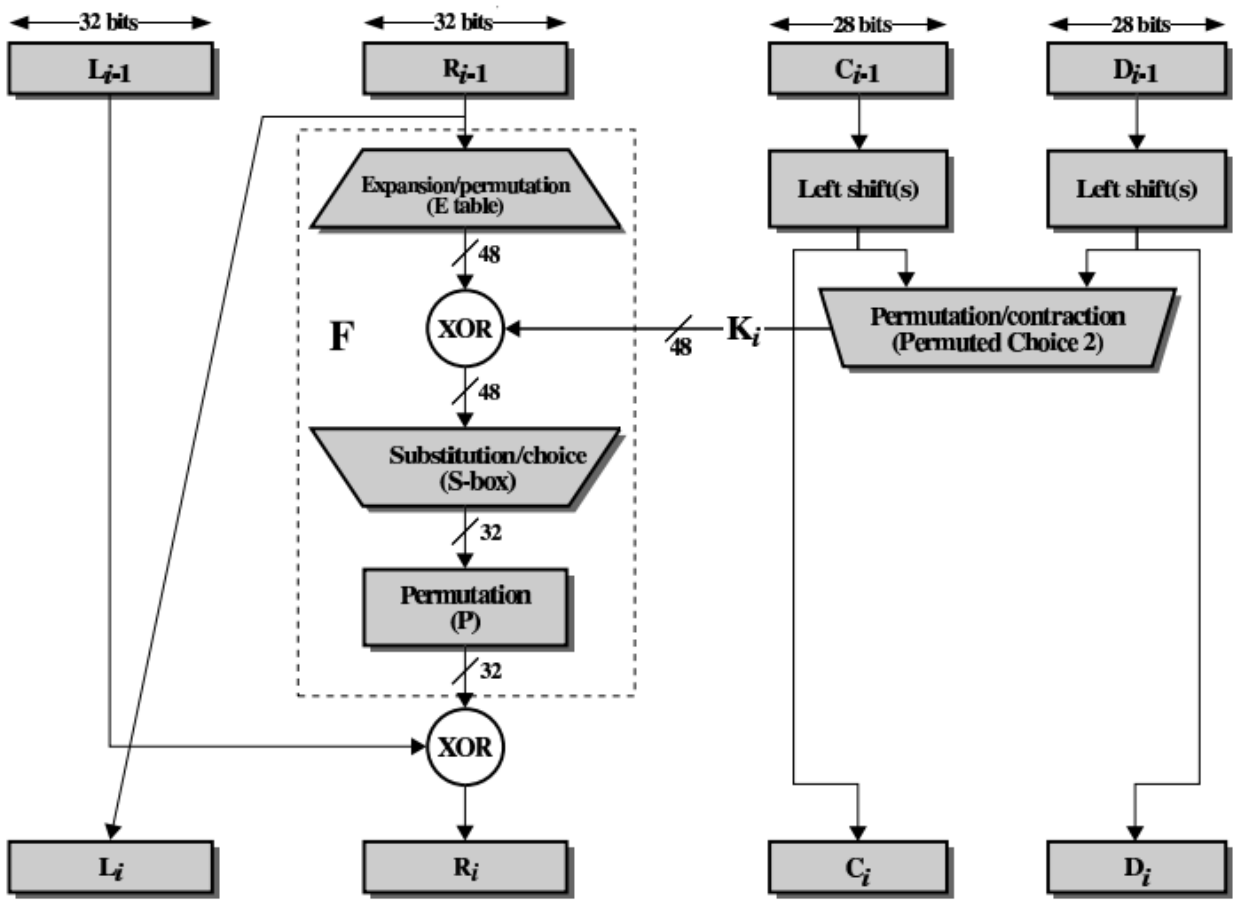
$$\begin{aligned} RD_{i+1} &= LD_i \oplus F(RD_i, K_{16-i}) \\ &= RE_{16-i} \oplus F(LE_{16-i}, K_{16-i}) \\ &= LE_{15-i} \oplus F(RE_{15-i}, K_{16-i}) \oplus F(RE_{15-i}, K_{16-i}) \\ &= LE_{16-(i+1)} \end{aligned}$$

باید توجه کنیم که ما هر تابعی را به عنوان F به کار نخواهیم برد. به خصوص، لازم نیست F وارون پذیر باشد.


Output (plaintext)
RD16 = LE0
LD16 = RE0
LD16 = RE0
RD16 = LE0
F
RD15 = LE1
LD15 = RE1
K1
F
LD14 = RE2
RD14 = LE2
K2
LD2 = RE14
RD2 = LE14
F
RD1 = LE15
LD1 = RE15
K15
F
LD0 = RE16
RD0 = LE16
K16
Input (ciphertext)


64-bit plaintext
64-bit key
Initial Permutation
Permuted Choice 1
Round 1
K1
Permuted Choice 2
Left circular shift
Round 2
K2
Permuted Choice 2
Left circular shift
Round 16
K16
Permuted Choice 2
Left circular shift
32-bit Swap
Inverse Initial Permutation
64-bit ciphertext


R (32 bits)
E
48 bits
K (48 bits)
+
S1
S2
S3
S4
S5
S6
S7
S8
P
32 bits



دو مطلب است که در مورد DES باید آنها را مورد تجزیه و تحلیل قرار داد:

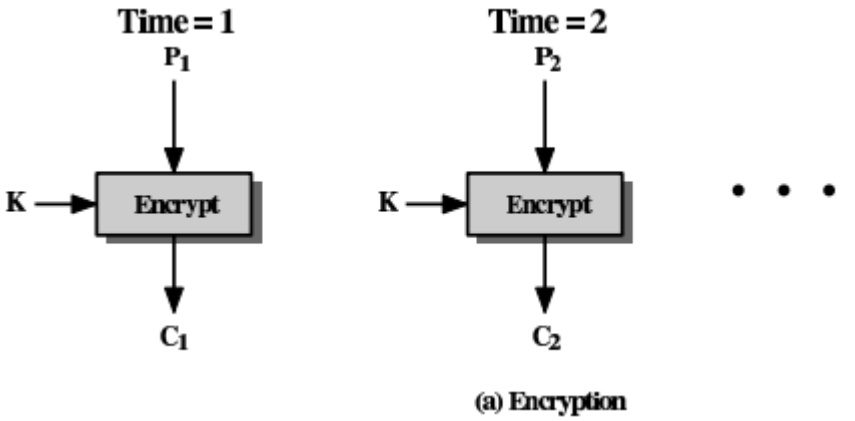
- S-boxها: تنها قسمت های غیرخطی بوده و برای ایجاد امنیت ضروری هستند. پیشنهاد شد که آنها شامل دریچه هایی هستند که به NSA اجازه می دهد تا بتواند DES را رمزگشایی کند. شواهدی که تاکنون جمع آوری شده نشان می دهد که S-boxها برای مقاومت در برابر برخی حملات پیشرفته خاص، مانند حمله تفاضلی، طراحی شده بودند، که برای NSA ۲۰ سال قبل شناخته شده بود، اما بعدها توسط بیهام و شامیر در سال ۱۹۹۱ مجدداً شناسایی شد. در ادامه خواهیم دید که حمله تفاضلی روی DES (با ۱۶ مرحله) نیازمند بررسی $2^{55.1}$ عملگر است که با 2^{56} عملگر مورد نیاز توسط حمله جستجوی جامع مقایسه می شود.
- اندازه کلید: لوسیفر اصلی ۱۲۸ بیت طول داشت؛ DES مطرح شده دارای ۶۴ بیت که به ۵۶ بیت کاهش می یافت (زیرا ۸ بیت آن بیت های توازن بودند)

- ۱۹۷۷: دیفر و هلمن یک ماشین ۲۰ ملیون دلاری را پیشنهاد دادند که قادر به شکستن DES در یک روز بود.
- ۱۹۹۳: وینر یک ماشین ۱۰۰ هزار دلاری را پیشنهاد داد که قادر به شکستن DES در ۱.۵ روز بود.
- ۱۹۹۸: یک ماشین ۲۵۰ هزار دلاری توسط موسسه EFC (Electronic Frontier Foundation) ساخته شد که توانست DES را در مدت زمان ۵۶ ساعت بشکند.
- ۱۹۹۹: یک حمله تحت اینترنت توانست DES را در مدت زمان ۲۲ ساعت و ۱۵ دقیقه بشکند.

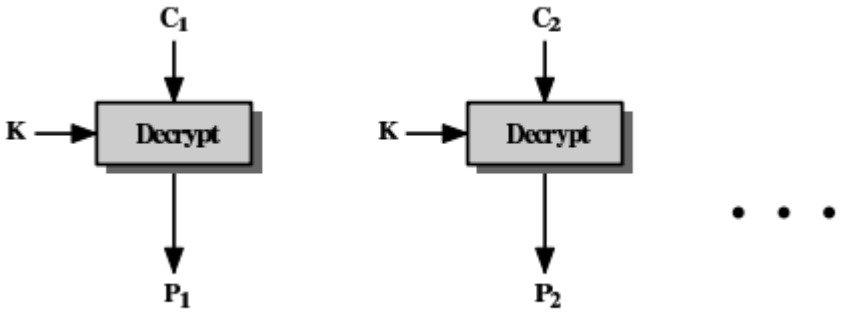
حمله خطی از حمله تفاضلی کاراتر است، DES با استفاده از 2^{43} زوج متن ساده-متن رمزشده شکسته شد. (البته، در عمل این تلاش خیلی کارامد نیست، چرا که نیازمند داشتن تعداد بسیار زیادی زوج متن ساده به همراه متن رمزشده متناظر با آنهاست).

۱- روش استفاده از کتابچه الکترونیکی (electronic codebook mode یا ECB):

- به ازای یک کلید داده شده، یک متن رمزشده یکتا برای هر ۶۴ بیت ورودی وجود دارد.
- برای پیام های کوتاه مانند کلید DES مناسب است.
- برای پیام های بلند (به خاطر منظم بودن آن) مناسب نیست.



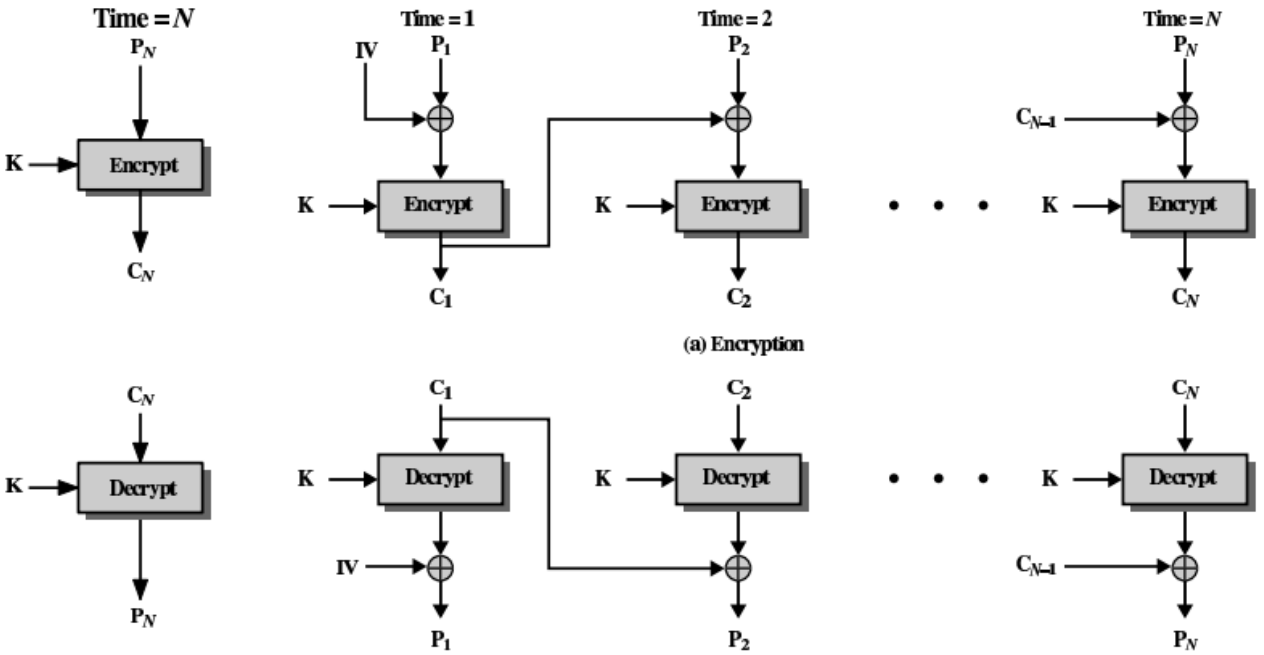
(a) Encryption



(b) Decryption

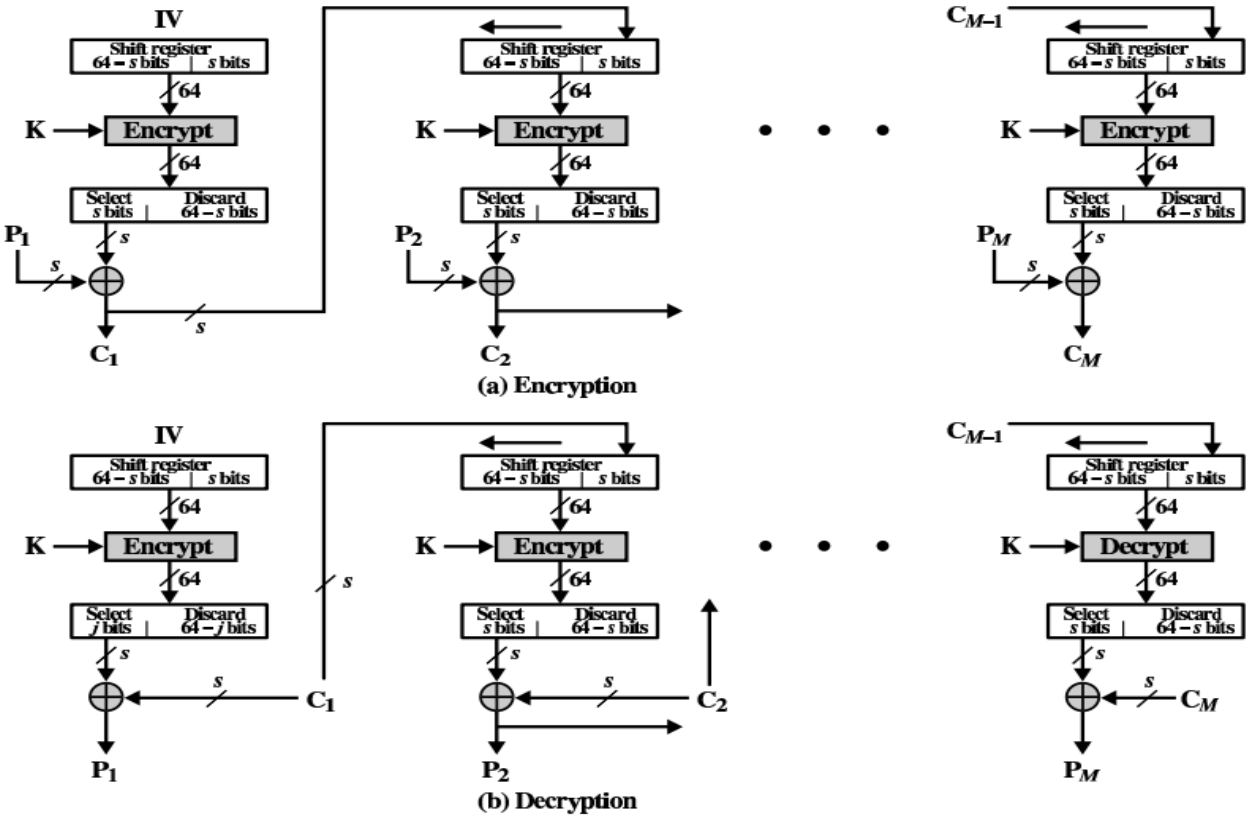
۲- روش زنجیره بلوکی رمز (CBC یا cipher block chaining mode):

- بلوک های یکسان از متن ساده، متن های رمزشده متفاوتی را تولید خواهند کرد.
- یک بردار اولیه IV برای اولین بلوک متن رمزشده به کار می رود؛ IV می بایست به صورت امن توسط دو طرف معلوم شده باشد؛ این بردار می تواند توسط روش ECB ارسال شود.
- اگر IV برای دشمن آشکار شود، آنگاه مشکلاتی ممکن است ظاهر شود؛ به طور مثال $C_1 = E_k(IV \oplus P_1)$ نتیجه می دهد $P_1 = IV \oplus D_k(C_1)$ ؛ بنابراین بیت های متناظر با P_1 و IV می توانند همزمان تکمیل شوند.



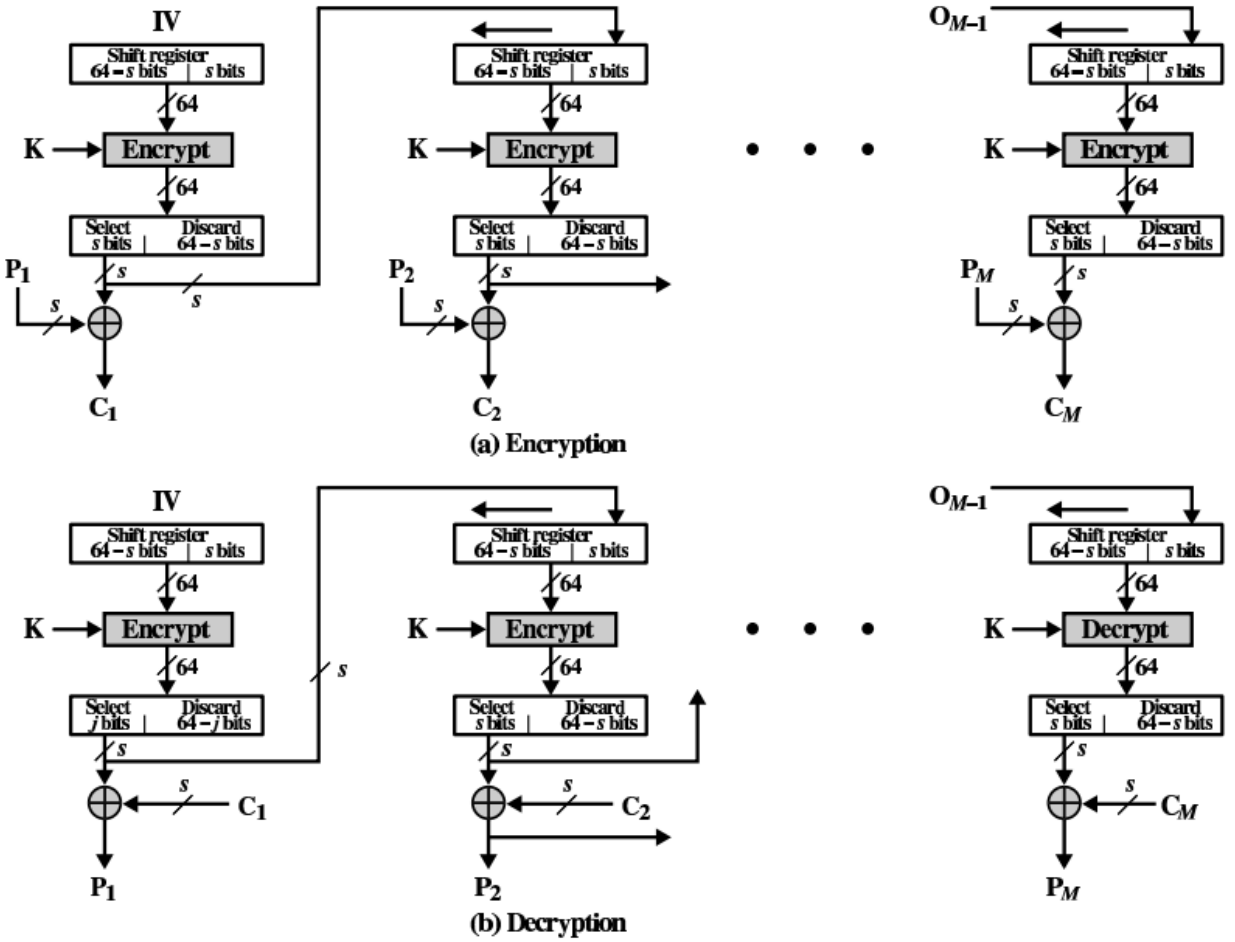
۳- روش بازخورد رو به عقب رمز (CFB یا cipher feedback mode):

- این روش، یک رمز دنباله ای است.
- متن رمزشده به عقب به رجیستر شیفتی بازخورد داده می شود.
- متن ساده به بلوک های S تایی تقسیم می شود.
- در زمان حقیقی محاسبات انجام می شود.
- برای تصدیق پیام مناسب است.
- به این مطلب توجه کنید که تنها از تابع رمزگذار استفاده شده است



4- روش بازخورد رو به عقب خروجی (output feedback mode یا OFB)

- به طور مشابه؛ خروجی رمزگذار به عقب به رجیستر باز می گردد-خطاهای بیتی در انتقال پخش نمی شوند (مورد استفاده برای انتقال ماهواره)
- در حمله تغییر رشته ورودی نسبت به CFB آسیب پذیرتر است.



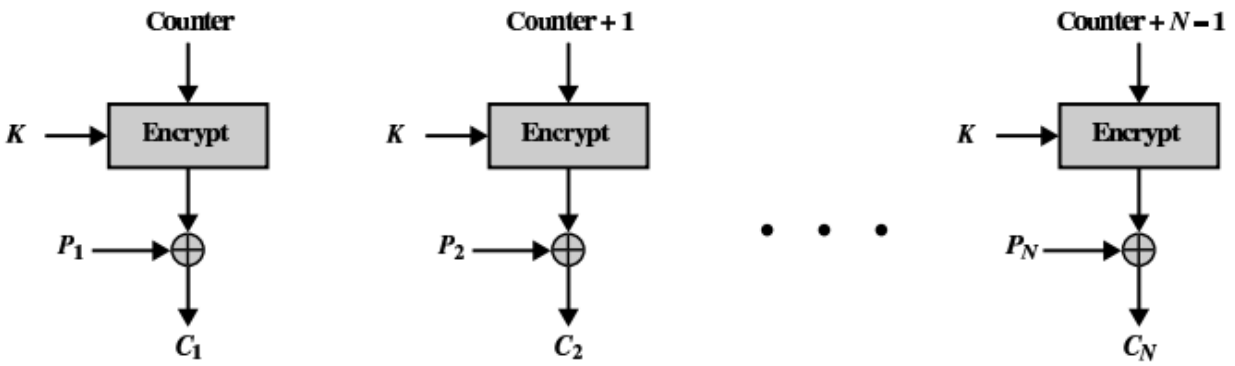
۵- روش شمارنده (counter mode یا CTR)

- این روش جدیدتر است.
- یک شمارنده به کار می رود؛ شمارنده باید برای هر بلوک رمزشده متفاوت باشد؛ معمولاً شمارنده در پیمانه اندازه بلوک به اندازه ۱ مقدار بیشتر می شود.
- مزایا:

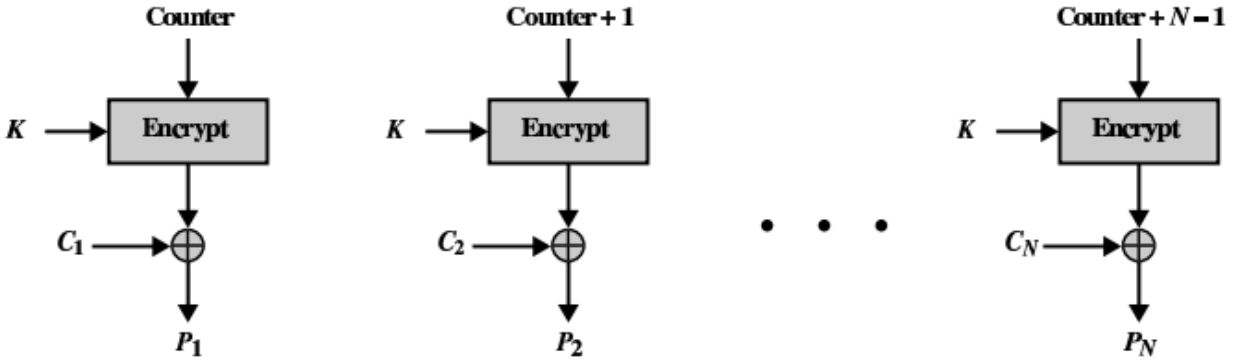
۱- کارآمد در سخت افزار و نرم افزار : می تواند به صورت موازی انجام شود.

۲- امکان دسترسی تصادفی به متن رمزشده.

۳- نیاز به تابع رمزگشای به کار رفته نیست.



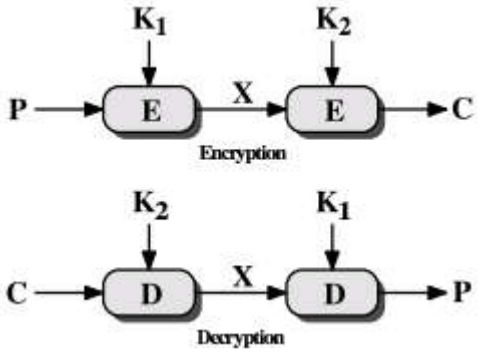
(a) Encryption



(b) Decryption

پس از مطرح شدن حملات تفاضلی و حملات خطی، دیگر سیستم DES امن نبود. این بود که روش دوتایی و سه تایی DES مطرح شد که به شکل زیر می باشند:

DES دوتایی (Double DES): دو کلید DES را به صورت زیر به کار می برد:

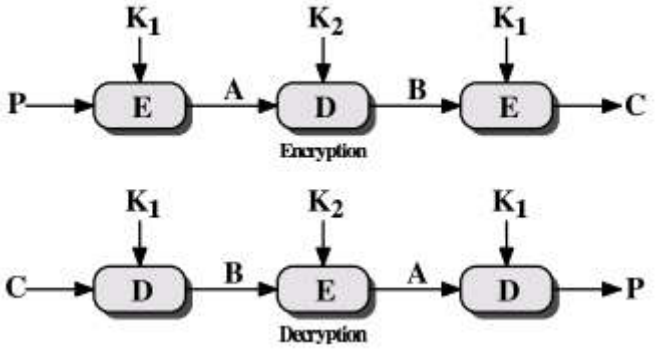


$$C = E_{K_2}(E_{K_1}(P)) \qquad P = D_{K_1}(D_{K_2}(C)).$$

به این مطلب باید توجه داشت که سیستم دوتایی DES را نمی توان با یک DES واحد شبیه سازی کرد، به این معنا که این سیستم کاملاً متفاوت عمل می کند. بنابراین، می بایست طول کلید به ۱۱۲ بیت افزایش یابد.

حمله مرد در میان (man-in-the-middle):

- داریم $E_{K_1}(P) = D_{K_2}(C)$.
- بنابراین، به ازای (P,C) داده شده، متن P را با به کارگیری تمامی 2^{56} مقدار برای کلید K_1 رمزگذاری کرده و آنها را در یک جدول ذخیره می کنیم.
- پس از آن، C را با به کارگیری تمامی 2^{56} رمز ممکن برای K_2 رمزگشایی کرده و آنها را با مقادیر داده شده در جدول قبل مطابقت می دهیم. (یافتن تطابق ها)
- هنگامی که یک تطابق رخ دهد، زوج کلید داده شده را برای یک زوج متن ساده-متن رمزشده متفاوت به کار می گیریم.
- هر متن ساده با استفاده از DES دوتایی به یکی از 2^{64} متن رمزشده ممکن رمزگذاری می شود؛ از آنجایی که 2^{112} کلید وجود دارد، به طور متوسط متن ساده P به یک متن رمزشده C با استفاده از 2^{48} کلید رمزگذاری می شود.
- بنابراین، برای اولین زوج، ایجاد یک تطابق با احتمال $1-2^{-48}$ با خطا مواجه می شود.
- پیغام خطا، برای هر دو زوج، با احتمال خیلی کم ($2^{-16} = 2^{48-64}$) ایجاد می شود.
- بنابراین، DES دوتایی خیلی نسبت به DES واحد امن نیست.
- **DES سه تایی (3DES):** مراحل سه تایی زیر را برای رمزگذاری و رمزگشایی با به کارگیری دو کلید K_1, K_2 دنبال می کند:
- $$C = E_{K_1}(D_{K_2}(E_{K_1}(P))) \qquad P = D_{K_1}(E_{K_2}(D_{K_1}(C)))$$



-

- تنها دلیل استفاده از تابع رمزگشا در وسط رابطه این است که به کاربران اجازه می دهد تا 3DES را برای رمزگشایی یک DES واحد به صورت زیر به کار برند:

$$C = E_K(P) = E_K(D_K(E_K(P))).$$

- -تاکنون، هیچ حمله شناخته شده موثری به 3DES صورت نگرفته است.
- - یک نفر ممکن است3DES را با سه کلید متفاوت به کار گیرد.
-
- تابع دوره ای و طراحی کلید:
- یک روش متداول که امروزه برای رمزهای بلوکی مدرن طراحی می شود، استفاده از رمز تکراری است. یک رمز تکراری شامل یک تابع دوره ای و یک طراحی کلید است. به ازای کلید داده شده K (معمولا یک کلید دوتایی تصادفی با طول مشخص است)، یک طراحی کلید $(K^1, K^2, ..., K^{N_r})$ را با استفاده از یک الگوریتم عمومی مشخص می سازیم؛ مولفه های K^r کلیدهای دوره ای نامیده می شوند. یک تابع دوره ای، مثلاً g، دو ورودی می گیرید: یک کلید دوره ای K^r و یک حالت جاری از متن ساده ای که رمزگذاری شده و حالت بعدی را تولید می کند. حالت اولیه، متن ساده بوده و حالت آخر، متن رمزشده است. بنابراین، الگوریتم رمزنگاری به صورت زیر است:

$$
\begin{aligned}
w^0 &\leftarrow x \\
w^1 &\leftarrow g(w^0, K^1) \\
w^2 &\leftarrow g(w^1, K^2) \\
&\vdots \quad \vdots \quad \vdots \\
w^{Nr-1} &\leftarrow g(w^{Nr-2}, K^{Nr-1}) \\
w^{Nr} &\leftarrow g(w^{Nr-1}, K^{Nr}) \\
y &\leftarrow w^{Nr}
\end{aligned}
$$

-

در مورد رمزهای بلوکی متقارن، حملات قدرتمند زیادی وجود دارد. در اینجا، به دو نمونه از آنها اشاره می کنیم. این حملات خیلی پیچیده بوده و ما تنها یک مدل ساده از آنها را که شبکه جایشگتی-جانیشینی (SPN) نام دارد، معرفی می کنیم.

شبکه جایگشتی-جانشینی (Substitution-permutation network) یا SPN.

یک شبکه SPN، حالت خاصی از یک رمز تکراری با تغییرات کمی است. فرض کنید l,m به عنوان دو عدد صحیح مثبت داده شده باشند (در اینجا lm طول بلوک رمز است). یک SPN از دو مولفه زیر تشکیل شده است:

۱- یک جانشینی (که معمولاً یک جایگشت است):

$$\pi_S : \{0,1\}^\ell \to \{0,1\}^\ell$$

۲- یک جایگشت:

$$\pi_P : \{1,2,\ldots,\ell m\} \to \{1,2,\ldots,\ell m\}.$$

π_S یک S-جعبه (S-box) نامیده می شود و یک l بیتی را با یک مجموعه l بیتی دیگر جایگزین می کند. π_P نیز برای جایگشت lm بیت به کار می رود.

به ازای رشته lm بیتی داده شده $x=(x_1,\ldots,x_{lm})$، x را به صورت یک اتصال از m تا زیررشته l بیتی $x_{(1)},x_{(2)},\ldots,x_{(m)}$ در نظر بگیرید. یعنی:

$$x = x_{(1)} \| x_{(2)} \| \cdots \| x_{(m)}$$

جایی که برای هر $1 \le i \le m$ ، داریم:

$$x_{(i)} = (x_{(i-1)\ell+1}, \ldots, x_{i\ell}).$$

شبکه جایگشتی-جانشینی (Substitution-permutation network) یا SPN.

$$P = C = \{0,1\}^{lm},\ K \subseteq (\{0,1\}^{lm})^{N_r+1}$$

رمزگذاری: N_r دوره که هر دوره (به جز آخری) شامل مراحل زیر است:

- Xor با یک کلید دوره ای (ادغام با کلید دوره ای)
- یک جانشینی با استفاده از π_S .
- یک جایگشت با استفاده از π_P .

$SPN(x, \pi_S, \pi_P, (K^1, K^2, \ldots, K^{Nr+1}))$

1. $w^0 \leftarrow x$
2. **for** r **from** 1 **to** $Nr-1$ **do**
3. $u^r \leftarrow w^{r-1} \oplus K^r$
4. **for** i **from** 1 **to** m **do**
5. $v^r_{(i)} \leftarrow \pi_S(u^r_{(i)})$
6. $w^r \leftarrow (v^r_{\pi_P(1)}, \ldots, v^r_{\pi_P(\ell m)})$
7. $u^{Nr} \leftarrow w^{Nr-1} \oplus K^{Nr}$
8. **for** i **from** 1 **to** m **do**
9. $v^{Nr}_{(i)} \leftarrow \pi_S(u^{Nr}_{(i)})$
10. $y \leftarrow v^{Nr} \oplus K^{Nr+1}$
11. **return** y

رمزگشایی: مشابه با رمزگذاری تنها با این تفاوت که:

- S-جعبه ها با مقادیر وارون آنها جایگزین می شوند:
- طراحی کلید به صورت برعکس صورت می گیرد.

- **مثال.** فرض کنید $l=m=Nr=4$ و π_S,π_P به صورت زیر تعریف شده باشند (در این تعریف، بردارهای دودویی به طول ۴ به صورت یک عدد در مبنای ۱۶ نمایش داده شده است):

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

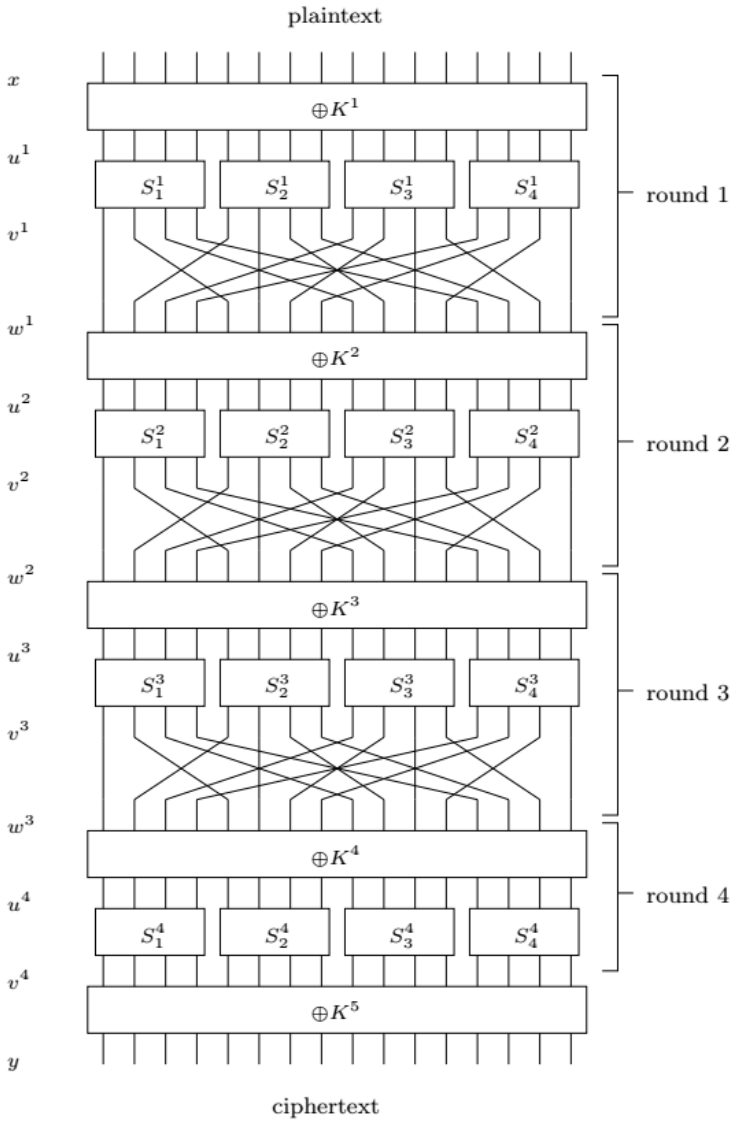
-
- این SPN در شکل سمت راست نیز نمایش داده شده است، جایی که S–جعبه ها دارای تعداد متفاوتی از جایگشت های ساده تر است. این جایگشت ها در کل، معرف یک S–جعبه خواهند بود.
- توصیف SPN ما با تخصیص الگوریتم تولید کلید کامل خواهد شد. برای این منظور یک مثال می آوریم. یک کلید ۳۲ بیتی $K=(k_1,\ldots,k_{32})\in\{0,1\}^{32}$ را در نظر بگیرید. برای $1\le r\le 5$، تعریف کنید K^r شامل ۱۶ بیت متوالی باشد که از k_{4r-3} شروع می شود. برای نمونه، اگر

$$K = 0011\ \ 1010\ \ 1001\ \ 0100\ \ 1101\ \ 0110\ \ 0011\ \ 1111.$$ -

- آنگاه کلید دورها به صورت زیر خواهد بود:

$$K^1 = 0011\ \ 1010\ \ 1001\ \ 0100$$
$$K^2 = 1010\ \ 1001\ \ 0100\ \ 1101$$
$$K^3 = 1001\ \ 0100\ \ 1101\ \ 0110$$
$$K^4 = 0100\ \ 1101\ \ 0110\ \ 0011$$
$$K^5 = 1101\ \ 0110\ \ 0011\ \ 1111$$ -

-



-

- در این حالت، اگر متن ساده به صورت زیر باشد:

$$x = 0010\ 0110\ 1011\ 0111$$

-

- آنگاه، متن رمزشده به صورت زیر به دست خواهد آمد:

$$
\begin{aligned}
w^0 &= 0010\ 0110\ 1011\ 0111 \\
K^1 &= 0011\ 1010\ 1001\ 0100 \\
u^1 &= 0001\ 1100\ 0010\ 0011 \\
v^1 &= 0100\ 0101\ 1101\ 0001 \\
w^1 &= 0010\ 1110\ 0000\ 0111 \\
K^2 &= 1010\ 1001\ 0100\ 1101 \\
u^2 &= 1000\ 0111\ 0100\ 1010 \\
v^2 &= 0011\ 1000\ 0010\ 0110 \\
w^2 &= 0100\ 0001\ 1011\ 1000 \\
K^3 &= 1001\ 0100\ 1101\ 0110 \\
u^3 &= 1101\ 0101\ 0110\ 1110 \\
v^3 &= 1001\ 1111\ 1011\ 0000 \\
w^3 &= 1110\ 0100\ 0110\ 1110 \\
K^4 &= 0100\ 1101\ 0110\ 0011 \\
u^4 &= 1010\ 1001\ 0000\ 1101 \\
v^4 &= 0110\ 1010\ 1110\ 1001 \\
K^5 &= 1101\ 0110\ 0011\ 1111 \\
y &= 1011\ 1100\ 1101\ 0110
\end{aligned}
$$

-

توضیحاتی درباره SPNها

- طراحی آنها ساده بوده و هم در سخت افزار و هم نرم افزار کارا می باشند.
- در نرم افزار، یک s–جعبه به صورت یک جدول مرجع مورد استفاده قرار می گیرد. حافظه مورد نیاز در آن $l2^l$ بیت است. در مثال قبل، هر s–جعبه نیازمند 2^6بیت است. AES – sجعبه ای را به کار می برد که ۸ بیت را به ۸ بیت تصویر می کند (اندازه کلید حداقل ۱۲۸ بیت، طول بلوک ۱۲۸ و حداقل ۱۰ روند داریم).
- در هر روند، یک نگاشت خطی وارون پذیر می تواند به عنوان یک جایگزین یا علاوه برآن، به عنوان یک عملگر جایگشتی، چنان که در AES به کار رفته است، بیاید.

حملات خطی و تفاضلی

این قسمت را با توصیف روش هایی برای شکستن رمز SPN آغاز می کنیم به طوری که این روش برای شکستن هر رمز تکراری به خوبی عمل می کند. فرض کنید توانسته ایم یک رابطه خطی

احتمالاتی بین زیرمجموعه ای از بیت های متن ساده و زیرمجموعه ای از بیت های حالت که دقیقاً قبل از عمل جایگشت در مرحله آخر وجود دارند، بیابیم. به عبارت دیگر، زیرمجموعه ای از بیت ها وجود دارند که عمل xor بین آنها به شکل غیرتصادفی عمل کرده است یا به زبان دیگر، احتمال آنها از ۱/۲ فاصله گرفته است. حال، فرض کنید حمله کننده به تعداد زیادی از زوج های شامل متن ساده-متن رمزشده دسترسی دارد که تمامی آنها با یک کلید نامشخص مانند K رمز شده اند (یعنی حمله متن ساده معلوم). برای هر یک از این زوج های شامل متن ساده-متن رمزشده، با استفاده از تمامی کلیدهای ممکن برای مرحله آخر، شروع به رمزگشایی می کنیم. متناظر با هر کلید، تمامی مقادیر بیت های حالت مربوطه که منجر به یک رابطه خطی می گردند را محاسبه و تعیین می کنیم که آیا رابطه خطی مذکور برقرار است یا خیر. زمانی که این رابطه برقرار شد، شماره گر مربوط به کلید داوطلب را به اندازه یک واحد افزایش می دهیم. در پایان فرآیند، امیدواریم که کلید داوطلب دارای فرکانس شمارشی باشد که از نصف تعداد زوج های شامل مقدار صحیح برای بیت های کلید به کاررفته دورتر باشد.

لم انباشتگی (*piling-up*)

فرض کنید X_i ها برای $i = 1, 2, \ldots$ متغیرهای تصادفی مستقلی باشند که دارای مقادیر 0,1 هستند و فرض کنید

$$\text{Prob}[X_i = 0] = p_i.$$

با توجه به استقلال X_i, X_j داریم:

$$\text{Prob}[X_i \oplus X_j = 0] = p_i p_j + (1-p_i)(1-p_j),$$
$$\text{Prob}[X_i \oplus X_j = 1] = p_i(1-p_j) + (1-p_i)p_j.$$

بایوس X_i به صورت زیر تعریف می شود:

$$\varepsilon_i = p_i - \frac{1}{2}$$

توجه داشته باشید $-1/2 \le \varepsilon_i \le 1/2$ ، $\text{Prob}[X_i = 0] = 1/2 + \varepsilon_i$ و $\text{Prob}[X_i = 1] = 1/2 - \varepsilon_i$.

لم (انباشتگی). برای $i_1 < i_2 < \cdots < i_k$ ، فرض کنید $\varepsilon_{i_1,i_2,\ldots,i_k}$ معرف بایاس متغیرهای تصادفی $X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_k}$ باشد.

در این صورت، داریم:

$$\varepsilon_{i_1,i_2,\ldots,i_k} = 2^{k-1} \prod_{j=1}^{k} \varepsilon_{i_j}.$$

نتیجه. اگر $\varepsilon_{i_1,i_2,\ldots,i_k}$ پایه متغیر تصادفی $X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_k}$ باشد و $\varepsilon_{i_j} = 0$ به ازای برخی مقادیر j، آنگاه $\varepsilon_{i_1,i_2,\ldots,i_k} = 0$.

لازم به ذکر است که لم پلنیگ-آپ در حالت کلی زمانی برقرار است که متغیرهای تصادفی مستقل باشند. به عنوان نمونه، متغیرهای مستقل X_1, X_2, X_3 با $\varepsilon_i = 1/4$ را برای تمامی iها در نظر بگیرید.

با توجه به لم پلینگ، داریم: $\varepsilon_{1,2} = \varepsilon_{1,3} = \varepsilon_{2,3} = 1/8$. با در نظرگرفتن $X_1 \oplus X_2$ و $X_2 \oplus X_3$ داریم:

$$(\mathrm{X}_1 \oplus \mathrm{X}_2) \oplus (\mathrm{X}_2 \oplus \mathrm{X}_3) = \mathrm{X}_1 \oplus \mathrm{X}_3$$

اگر $X_1 \oplus X_2$ و $X_2 \oplus X_3$ مستقل باشند، خواهیم داشت $\varepsilon_{1,3} = 2(1/8)2 = 1/32$. اما $\varepsilon_{1,3} = 1/8$.

تقریب خطی S-جعبه ها.

یک S-جعبه $\pi_S : \{0,1\}^m \to \{0,1\}^n$ را در نظر بگیرید؛ توجه داشته باشید که نیاز نداریم که $m = n$. یک ورودی به صورت $X = (X_1, \ldots, X_m)$ است، جایی که هر x_i، متغیر تصادفی X_i را تعریف می‌کند که مقادیر ۰ و ۱ را اختیار می‌کند و دارای بایاس $\varepsilon_i = 0$ است و این متغیرها مستقل هستند.

خروجی به صورت $Y = (y_1, \ldots, y_n)$ است و هر y_i یک متغیر Y_i را تعریف می‌کند. به وضوح، این متغیرها از یکدیگر ونیز از X_iها مستقل نمی باشند.

پس از آن، بایاس متغیرهای به شکل زیر را محاسبه می کنیم.

$$X_{i_1} \oplus \cdots \oplus X_{i_k} \oplus Y_{j_1} \oplus \cdots \oplus Y_{j_\ell}.$$

یک حمله تحلیل رمز خطی می تواند به طور بالقوه صورت گیرد، زمانی که یک متغیرتصادفی به این شکل دارای بایاسی است که از صفر فاصله گرفته است.

مثال. برای S-box موجود در مثال قبل، تمامی مقادیر ممکن اختیار شده توسط هشت متغیر تصادفی $X_1, \ldots, X_4, Y_1, \ldots, Y_4$ در جدول زیر محاسبه شده است:

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_1 \oplus X_4 \oplus Y_2$	$X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$
0	0	0	0	1	1	1	0	1	1
0	0	0	1	0	1	0	0	0	1
0	0	1	0	1	1	0	1	1	1
0	0	1	1	0	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0
0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	0	1	1	0	1
0	1	1	1	1	0	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	0	0	1	1	0	1	0	0	0
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	1	0	0	1	1
1	1	0	0	0	1	0	1	0	1
1	1	0	1	1	0	0	1	0	1
1	1	1	0	0	0	0	0	1	1
1	1	1	1	0	1	1	1	1	1

حال، اگر ما متغیرتصادفی $X_1 \oplus X_4 \oplus Y_2$ را در نظر بگیریم، بایاس این متغیر برابر با صفر است (همچنان که در جدول فوق دیده می شود). بنابراین، این متغیر برای حمله خطی، مناسب نیست. از طرف دیگر، متغیرتصادفی $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$ دارای بایاس $-\frac{3}{8}$ است، آنچنان که در جدول فوق دیده می شود.

حال، بایاس های تمامی $2^8 = 256$ متغیرتصادفی به این شکل را محاسبه می کنیم. ما هر یک از چنین متغیرهای تصادفی را به شکل زیر نمایش می دهیم:

$$(\bigoplus_{i=1}^{4} a_i X_i) \oplus (\bigoplus_{i=1}^{4} b_i Y_i)$$

جایی که $a_i, b_i \in \{0,1\}$. سپس، ما هر یک از ۴-تایی های $a = (a_1, a_2, a_3, a_4)$ و $b = (b_1, b_2, b_3, b_4)$ را به صورت اعدادی در مبنای ۱۶ در نظر می گیریم؛ جمعوندهای بالا "جمعوند داخلی" و "جمعوند خارجی" نامیده می شود. $N_L(a, b)$ را تعداد ۸-تایی های دودویی $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$ تعریف کنید، به طوری که:

$$\pi_s(x_1, x_2, x_3, x_4) = (y_1, y_2, y_3, y_4)$$

و

$$(\overset{4}{\underset{i=1}{\oplus}} a_i x_i) \oplus (\overset{4}{\underset{i=1}{\oplus}} b_i y_i) = 0$$

توجه داشته باشید که بایاس متغیرتصادفی که دارای مجموع ورودی a و مجموع خروجی b است، به صورت زیر می باشد:

$$\varepsilon(a, b) = \frac{N_L(a, b) - 8}{16}$$

جدول شامل تمامی مقادیر N_L جدول تقریب خطی (linear approximation table) نامیده می شود. برای مثال، این جدول در شکل زیر نمایش داده شده است.

$N_L(a,b)$	b (output sum)															
a (input sum)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	8	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8

یک حمله خطی روی SPN .

حملات خطی، نیازمند یافتن مجموعه ای از تقریب های خطی S- جعبه هاست که می تواند به کار رفته تا یک تقریب خطی روی تمام SPN (غیر از مرحله آخر) را نتیجه دهد.

ما این فرآیند را با استفاده از SPN موجود در مثال قبل، تشریح می کنیم. نیز، این حمله در شکل زیر نمایش داده شده است که در آن خطوط ضخیم متناظر با متغیرهای تصادفی هستند که در آن تقریب خطی شرکت کرده اند و S–جعبه های برچسب گذاری شده آنهایی هستند که در این تقریب ها حضور دارند-آنها S–جعبه های فعال نامیده می شوند.

این تقریب، چهار تا S–جعبه فعال را با هم ترکیب می کند:

۱-در S_2^1: متغیرتصادفی $T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$ دارای بایاس ۱/۴ است.

۲-در S_2^2: متغیرتصادفی $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$ دارای بایاس ۱/۴- است.

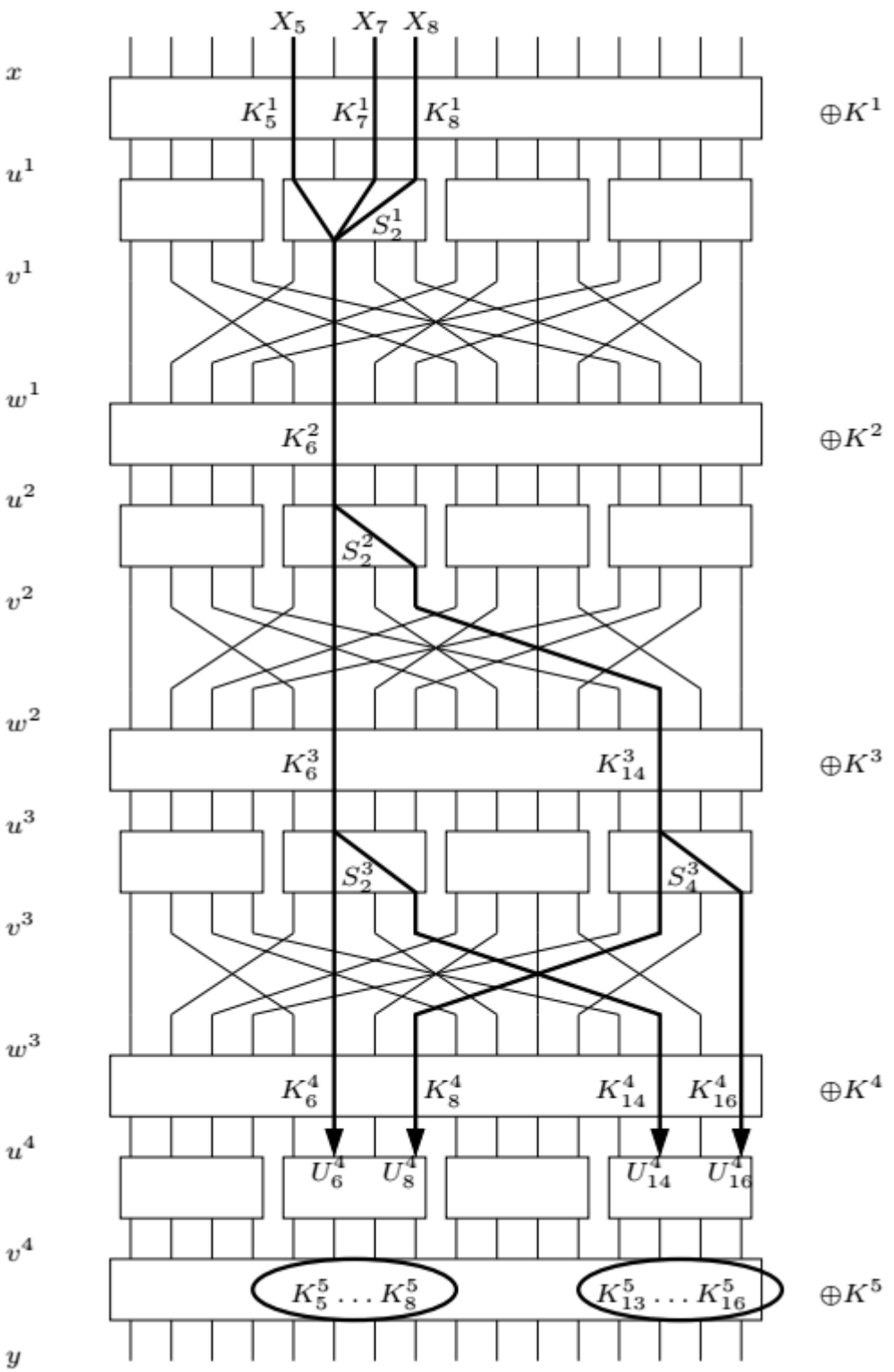
۳-در S_2^3: متغیرتصادفی $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$ دارای بایاس ۱/۴- است.

۴-در S_4^3: متغیرتصادفی $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$ دارای بایاس ۱/۴- است.

چهار متغیر تصادفی T_i دارای بایاس هایی هستند که مقدار مطلق آنها زیاد است. نیز، xor آنها منجر حذف متغیرهای تصادفی میانی می شود. اگر ما فرض کنیم که این چهار متغیر تصادفی مستقل هستند، آنگاه می توانیم بایاس xor آنها را

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4$$

با استفاده از لم پلینگ محاسبه کنیم. (در واقع، این متغیرها مستقل نیستند، که به این معناست که لم پلینگ نتیجه صحیح را نخواهد داد. اما، در عمل، تقریب نسبتاً خوبی را می دهد که به خوبی برای چهار حمله ما کار می کند.) بنابراین، با استفاده از لم پلینگ، فرض می کنیم که متغیرتصادفی $T_1 \oplus T_2 \oplus T_3 \oplus T_4$ دارای بایاس ۱/۳۲- است.



پس از آن، با توجه به شکل بالا می توانیم بنویسیم:

$$T_1 = X_5 \oplus K_5^1 \oplus X_7 \oplus K_7^1 \oplus X_8 \oplus K_8^1 \oplus V_6^1$$
$$T_2 = V_6^1 \oplus K_6^2 \oplus V_6^2 \oplus V_8^2$$
$$T_3 = V_6^2 \oplus K_6^3 \oplus U_6^4 \oplus K_6^4 \oplus U_{14}^4 \oplus K_{14}^4$$
$$T_4 = V_8^2 \oplus K_{14}^3 \oplus U_8^4 \oplus K_8^4 \oplus U_{16}^4 \oplus K_{16}^4$$

با xor نمودن $T_1 \oplus T_2 \oplus T_3 \oplus T_4$ مقدار زیر را داریم:

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$$

بنابراین، آخرین متغیر تصادفی نیز دارای بایاس (تقریباً) ۱/۳۲- است. این مقدار تنها شامل بیت های متن ساده، از u^4 و کلید است. فرض کنید بیت های کلید ثابت باشند. در این صورت، متغیرتصادفی:

$$\mathrm{K}_1^5 \oplus \mathrm{K}_7^1 \oplus \mathrm{K}_8^1 \oplus \mathrm{K}_6^2 \oplus \mathrm{K}_6^3 \oplus \mathrm{K}_{14}^3 \oplus \mathrm{K}_6^4 \oplus \mathrm{K}_8^4 \oplus \mathrm{K}_{14}^4 \oplus \mathrm{K}_{16}^4$$

دارای مقدار ثابت ۰ یا ۱ است. بنابراین، متغیرتصادفی

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4$$

بسته به مقادیر بیت های کلید، دارای بایاس (تقریباً) ۱/۳۲± است. این بایاس، به ما این اجازه را می‌دهد که حمله خطی را صورت دهیم.

فرض کنید ما دارای N_l زوج متن ساده-متن رمزشده هستیم که همگی از کلید نامشخص و یکسان K استفاده می‌کنند. این حمله، به ما اجازه می دهد تا بیت های کلید زیر را به دست آوریم:

$$K_5^5,\ K_6^5,\ K_7^5,\ K_8^5,\ K_{13}^5,\ K_{14}^5,\ K_{15}^5,\ K_{16}^5,$$

یعنی، هشت بیت کلید که با خروجی S–جعبه های S_2^4 و S_4^4 xor شده اند (آنها متناظر با بیت هایی از u^4 هستند که در ۴ رابطه خطی مان شرکت دارند). به ازای مقادیری که این هشت بیت می‌گیرند، $2^8 = 256$ احتمال خواهیم داشت. هر یک از این ۸ تایی‌ها شامل مقادیری برای این هشت بیت کلید است که زیرکلید کاندید (candidate subkey) نامیده می شود.

برای هر زوج (x, y) از متن‌ساده-متن‌رمزشده و هر زیرکلیدکاندید، یک رمزگشایی جزیی از yرا انجام داده تا به مقادیر حاصل برای $u_{(2)}^4$ و $u_{(4)}^4$ دست یابیم. در این صورت، ما مقدار

$$x_5 \oplus x_7 \oplus x_8 \oplus u_4^6 \oplus u_4^8 \oplus u_4^{14} \oplus u_4^{16}.$$

ما، آرایه‌ای از شمارنده ها که با ۲۵۶ زیرکلید کاندید اندیس‌گذاری شده را نگهداری کرده و شمارنده متناظر با یک زیرکلید خاص را در حالی که مقدار قبلی برابر با صفر است، افزایش می‌دهیم.

در پایان، انتظار داریم که اکثر شمارنده ها نزدیک به $N_l/2$ باشند، اما شمارنده متناظر با کلید کاندید صحیح نزدیک به $N_l/2 \pm N_l/32$ خواهد بود. این مطلب به طور امیدوارانه‌ای به ما اجازه می‌دهد تا زیرکلید صحیح را مشخص کنیم.

برای مثال خودمان، برخی نتایج جزیی برای شمارنده‌های متناظر با زیرکلیدکاندید در جدول زیر نشان داده شده است: در این جدول، داریم $N_l = 10000$ و $|bias| = |count - 5000|/10000$.

توجه داشته باشید که مقدار متناظر با زیرکلید $(2,4)_{hex}$ دارای مقدار متناظر 0.0336 است که به مقدار مورد نظر $1/32 = 0.03125$ بسیار نزدیک است.

candidate subkey $(K_5^5, \ldots, K_8^5, K_{13}^5, \ldots, K_{16}^5)$	|bias|
1 C	0.0031
1 D	0.0078
1 E	0.0071
1 F	0.0170
2 0	0.0025
2 1	0.0220
2 2	0.0211
2 3	0.0064
2 4	**0.0336**
2 5	0.0106
2 6	0.0096
2 7	0.0074
2 8	0.0224
2 9	0.0054
2 A	0.0044
2 B	0.0186
2 C	0.0094

پیچیدگی حمله.

فرض کنید ε معرف بایاس این احتمال باشد که یک توصیف خطی برای رمز کامل برقرار باشد. تعداد N_l زوج متن ساده-متن رمزشده معلوم مورد نیاز تقریباً به صورت زیر است:

$$N_l \approx 1/\varepsilon^2.$$

در عمل، N_l یک ضریب کوچک از $1/\varepsilon^2$ است. در مثال ما، N_l تقریباً ۱۰ برابر ε^2 بود.