Shahid Beheshti University

رمزنگاری

هادی سلیمانی

پژوهشکده فضای مجازی دانشگاه شهید بهشتی

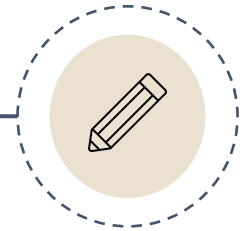
مجوز استفاده و نشر

- اجازه‌ی ایجاد نسخه‌های دیجیتالی جدید براساس بخشی یا تمام مطالب این اسلاید بدون پرداخت هزینه اعطا می‌شود، مشروط بر این‌که:
  - فقط به‌منظور و در راستای استفاده‌ی آموزشی (شخصی و یا کلاسی) ساخته شده باشند و برای کسب هرگونه سود و یا مزیت تجاری استفاده نشوند.
  - نسخه‌های جدید حاوی ارجاع مستقیم به نام تهیه‌کننده اسلاید (هادی سلیمانی) و محل کار وی (پژوهشکده فضای مجازی دانشگاه شهید بهشتی) باشند.
- مجموعه‌ی حاضر براساس نظرات ارزشمند دانشجویان (سابق) دانشگاه شهید بهشتی و همکاران محترم تهیه شده است که از تمام آن‌ها قدردانی می‌شود؛

  (به‌خصوص خانم‌ها **سارا زارعی** و **فاطمه عزیزی** نقش مهمی را در تهیه نسخه‌ی نهایی بر عهده داشته‌اند. خانم مهندس زارعی علاوه بر کمک در آماده‌سازی نسخه‌ی فعلی اسلایدها، در تصحیح اشتباهات نسخه‌ی قبلی و همچنین تکمیل و بازتعریف محتوای درس‌ها بسیار تاثیرگذار بوده‌اند).
- برای مشاهده‌ی اسلایدها و ویدئوهای تدریس این درس به آدرس زیر مراجعه فرمایید:

http://facultymembers.sbu.ac.ir/h_soleimany/cryptography-course/

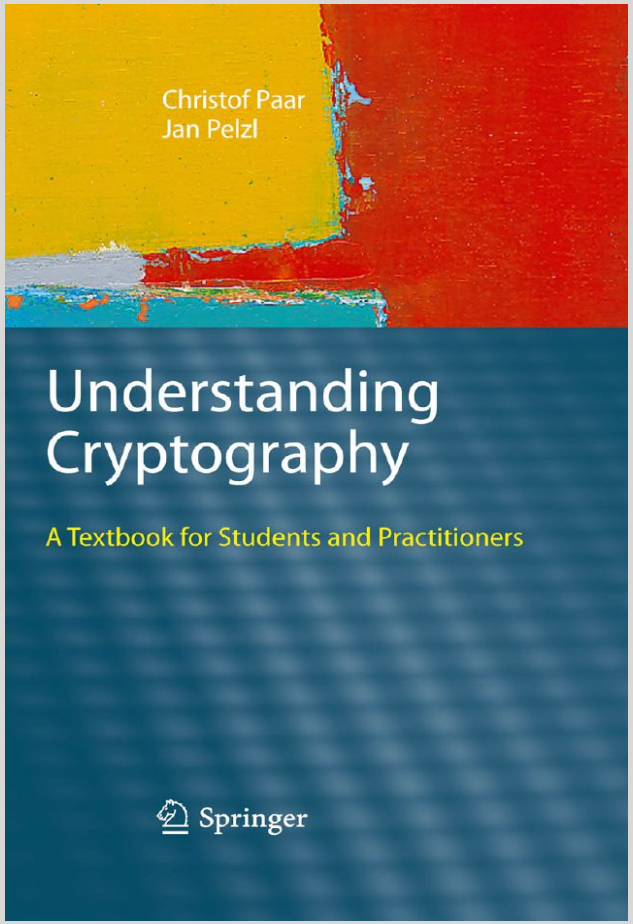
درس سوم



مقدمه‌ای بر سیستم‌های رمزنگاری کلاسیک

معرفی مـرجع

مقدمه‌ای بر سیستم‌های رمزنگاری کلاسیک



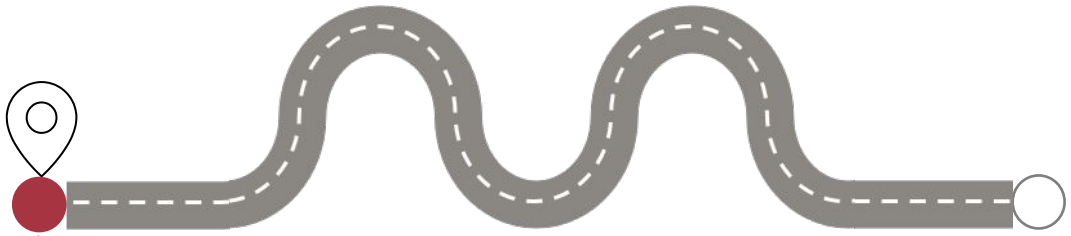
Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.

مثال‌ها و تصاویر این بخش از درس از کتاب مرجع است (در غیر این صورت مرجع شکل ذکر شده است).

فهرست عناوین درس

مقدمه‌ای بر سیستم‌های رمزنگاری کلاسیک

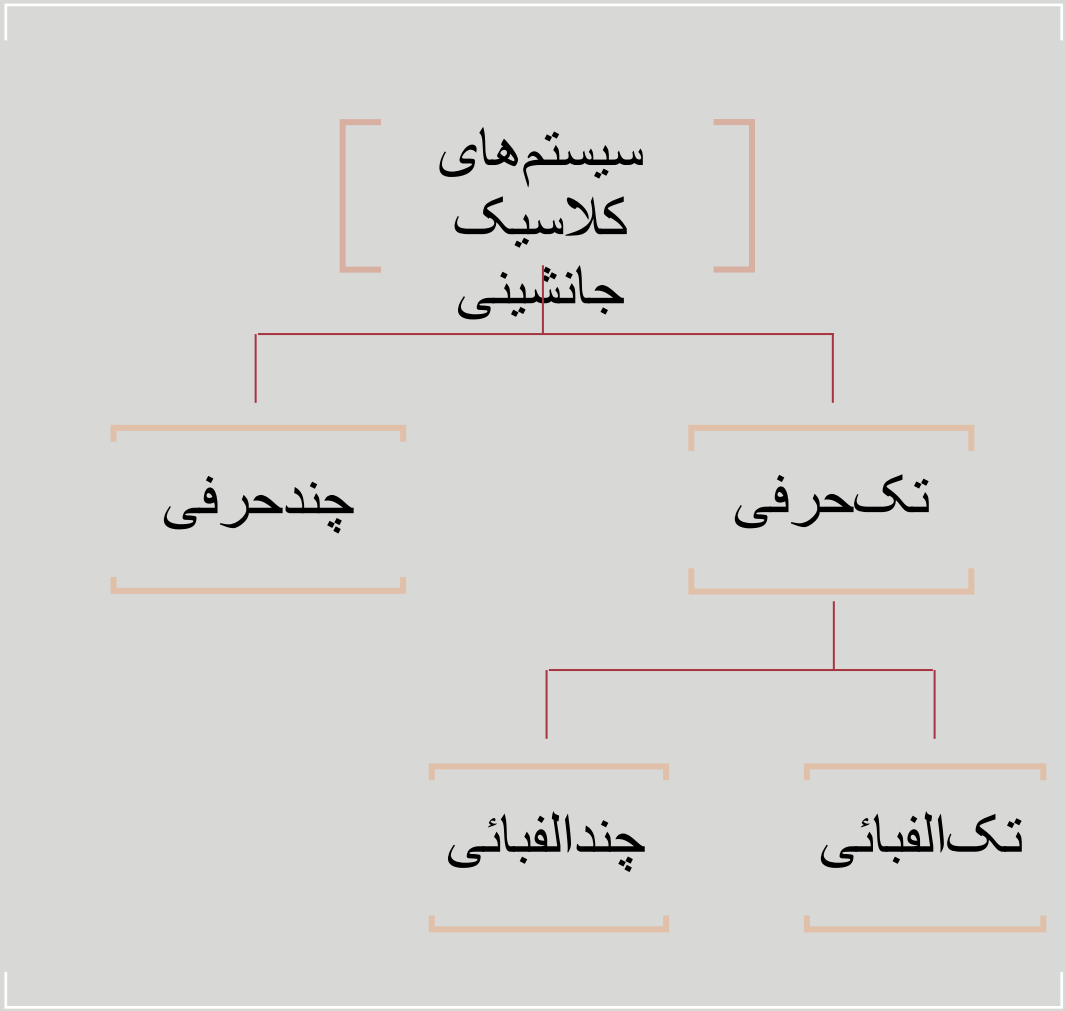
- مقدمه
- رمزنگاری تک‌حرف و تک‌الفبا
- رمزنگاری تک‌حرف و چندالفبا
- رمزنگاری چندحرف
- جمع‌بندی مطالب



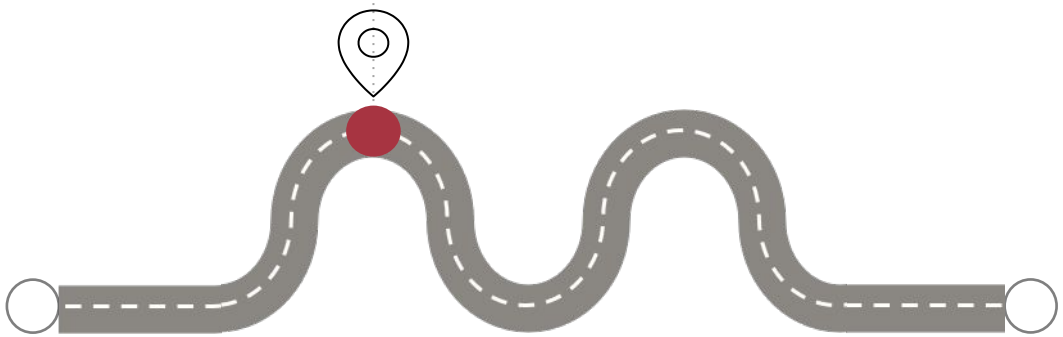
مقدمه

مقدمه

- پیش از اختراع و همه‌گیر شدن استفاده از رایانه، سیستم‌های رمزنگاری مبتنی بر رمزنگاری حروف بودند.
- به طور معمول در متون علمی از این دسته از رمزها، با عنوان سیستم‌های رمزنگاری کلاسیک یا غیرمدرن نام می‌برند.
- سیستم‌های کلاسیک عموما حروف متن اصلی را با حروف دیگر جایگزین می‌کنند.
- در هر بار جایگزینی، یک یا چند حرف رمز می‌شود.
- اگر معادل رمزشده‌ی یک حرف ثابت، همیشه یکسان باشد به آن تک‌الفبائی و در غیراین‌صورت چندالفبائی می‌گویند.



رمزنگاری
تک‌حرف و
تک‌الفبا



مقدمه

هم‌نهشتی

(یادآوری)

- برای اعداد صحیح a، b و m، می‌گوییم عدد a به پیمانه (یا هنگ) m با b هم‌نهشت است اگر $m|(a-b)$ و می‌نویسیم $a \equiv b \bmod (m)$.
- عدد m را پیمانه می نامند.
- مثال:

$$12 \equiv 3 \; mod \; 9$$
$$34 \equiv 7 \; mod \; 9$$
$$-7 \equiv 2 \; mod \; 9$$

ویژگی‌های محاسبات پیمانه‌ای

- عدد a به پیمانه‌ی m با تعداد بی‌شماری عدد هم‌نهشت است (و نه فقط با یک عدد خاص):

$$12 \equiv 3 \equiv 21 \equiv -6 \mod 9$$

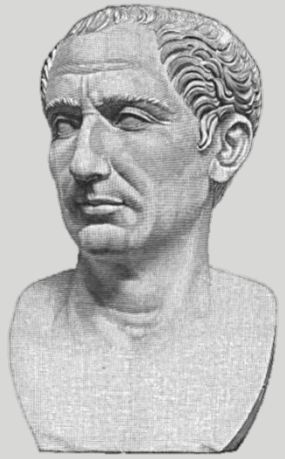
- به منظور سادگی، معمولا در محاسبات پیمانه‌ای، برای یک عدد صحیح دلخواه a به پیمانه‌ی m، کوچک‌ترین عدد طبیعی r را در نظر می‌گیریم که:

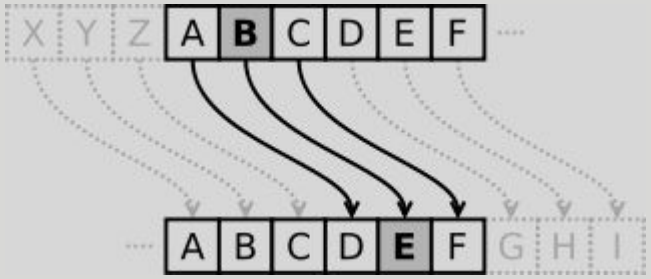
$$a = q.m + r$$

- مثلا برای مثال قبلی ($a = 12$ و پیمانه $m = 9$)، هم‌نهشتی زیر را در نظر می‌گیریم:

$$12 \equiv 3 \mod 9$$

Shift Cipher





- فضای متن اصلی، حروف الفبای موردنظر است که به ترتیب با یک عدد بین 0 تا $n-1$ نمایش داده می‌شوند و n تعداد حروف آن الفبا است.
- مثلا اگر الفبای انگلیسی را در نظر بگیریم:

$$\{A, \dots, Z\} \approx \{0, \dots, 25\}$$

- کلید k نیز عددی بین بین 0 تا $n-1$ است.
- رمزگذاری iامین حرف (m_i):

$$Enc_k(m_i) = m_i + k \bmod 26$$

- رمزگشایی iامین حرف (c_i):

$$Dec_k(c_i) = c_i - k \bmod 26$$

- ژولیوس سزار از شخصیت‌ها تاریخی روم (۱۰۰ سال قبل از میلاد مسیح) از این رمز با کلید $k = 3$ استفاده می‌کرده است.

Figures source: Wikipedia

مثال

- برای رمز کردن کلمه EXAMPLE تحت $k = 15$، داریم:

EXAMPLE = 4,23,0,12,15,11,4

$$c_i = m_i + 15 \bmod 26$$

19,12,15,1,4,0,19 = TMPBEAT

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

امنیت الگوریتم رمزنگاری Shift Cipher

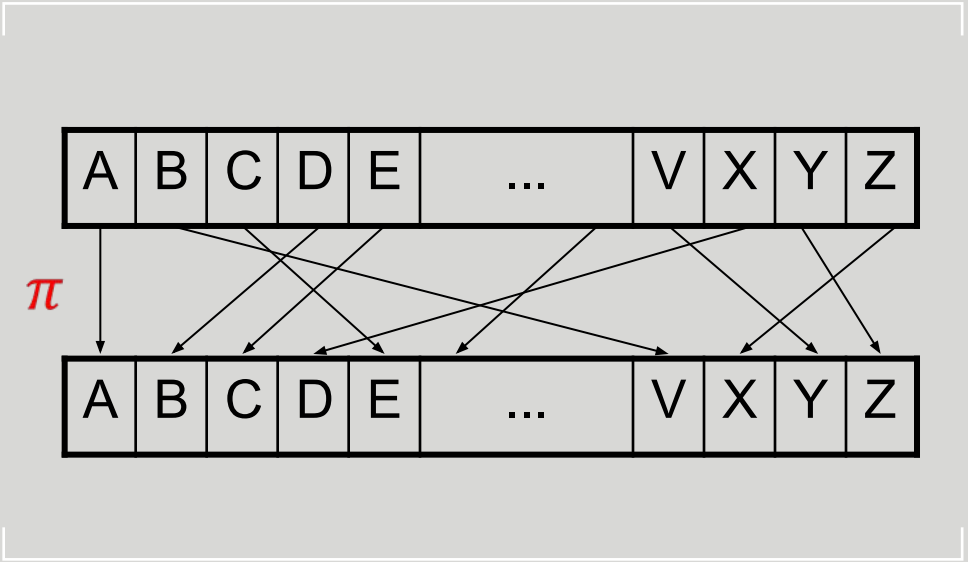
- چگونه می‌توان الگوریتم رمزنگاری Shift Cipher را شکست؟

الگوریتم حمله:

- فرض کنید که متن رمزشده‌ی C داده شده باشد.
- برای تمام کاندیدهای کلید $k \in \{0, \dots, 25\}$ عمل رمزگشایی را انجام داده و چک می‌کنیم که آیا متن معادل اصلی به دست آمده معنی‌دار هست یا خیر؟
- به احتمال بسیار زیاد تنها کلید صحیح منجر به یک متن معنی‌دار خواهد شد.

- به این حمله اصطلاحاً جست‌وجوی کامل (Brute Force یا Exhaustive Search) می‌گویند.
- حمله‌ی جست‌وجوی کامل به هر الگوریتمی قابل اعمال می‌باشد.
- تنها راه‌کار مقابله با این حمله، افزایش فضای کلید است.

Substitution Cipher



- فضای متن اصلی و متن رمزشده $\{0, \dots, n-1\}$ است که n تعداد حروف زبان است: $\{A, \dots, Z\} \approx \{0, \dots, 25\}$
- فضای کلید، مجموعه‌ی تمامی جایگشت‌های ممکن برای مجموعه‌ی $\{A, \dots, Z\} \approx \{0, \dots, 25\}$ است.
- رمزگذاری iامین حرف (m_i):

$$Enc_k(m_i) = \pi(m_i)$$

- رمزگشایی iامین حرف (c_i):

$$Dec_k(c_i) = \pi^{-1}(c_i)$$

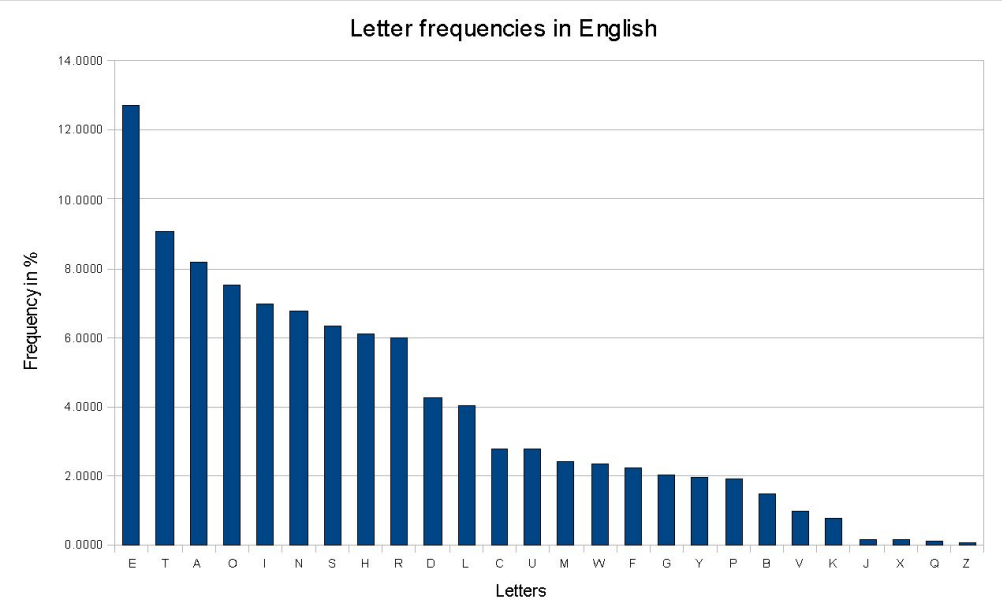
- تعداد حالات ممکن برای کلید:

$$26 \times 25 \times \cdots \times 1 = 26! \approx 2\text{^}88$$

- جست‌وجوی کامل به لحاظ محاسباتی امکان‌پذیر نیست.
- آیا می شود نتیجه گرفت که این رمز امن است؟

جدول فرکانسی

(Frequency Table)



- می‌توان میزان کاربرد هر کدام از حروف زبان را محاسبه و در یک جدول به نام جدول فرکانسی (Frequency Table) ذخیره کرد.
- برای مثال در زبان انگلیسی حرف E بیش‌ترین کاربرد و حرف Z کم‌ترین کاربرد را دارند.
- مشاهده‌ی اصلی این است که توزیع موجود در متن اصلی کماکان در متن رمزشده متفاوت و تنها جای حروف عوض شده است.
- از این طریق ممکن است که بتوان نحوه جابه‌جایی را تشخیص داد.

جدول فرکانسی

مثال

- متن رمزشده‌ی زیر را در نظر بگیرید:

```
iq ifcc vqqr fb rdq vfllcq na rdq cfjwhwz hr bnnb hcc
hwwhbsqvqbre hwq vhlq
```

- با توجه به خواص آماری می‌توان گفت که احتمالا معادل متن اصلی q، حرف e است.

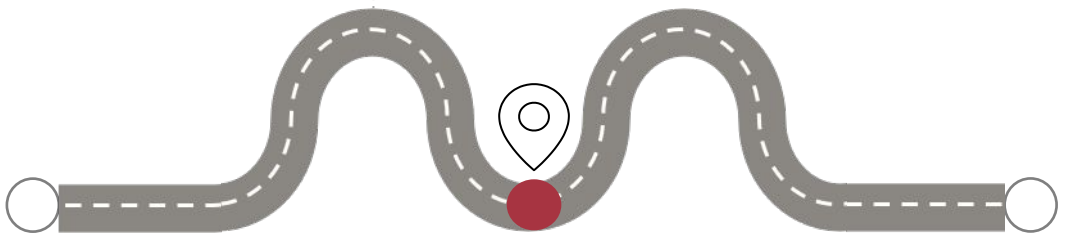
```
iE ifcc vEEr fb rdE vfllcE na rdE cfjwhwz hr bnnb  hcc
hwwhbsEvEbre hwE vhlE
```

- با روش مشابه و با سعی و خطا می‌توان متن اصلی را به صورت زیر بازیابی کرد:

```
WE WILL MEET IN THE MIDDLE OF THE LIBRARY AT NOON ALL
ARRANGEMENTS ARE MADE
```

Example's source: Understanding Cryptography: A Textbook for Students and Practitioners By Christof Paar, Jan Pelzl, page 8

رمزنگاری
تک‌حرف و
تک‌الفبا



مقدمه

**رمزنگاری
تک‌حرف و
چندالفبا**

Vigenère Cipher

- کلید به صورت $K = (k_1, k_2, \dots, k_\ell)$ تعریف می‌شود که هر کدام از k_iها عددی بین 0 تا $n - 1$ است (که n تعداد حروف الفبای موردنظر است).
- متن اصلی به دسته‌های ℓ حرفی تقسیم می‌شود.
- عملیات رمزگذاری ℓ حرف $(m_1, \dots, m_\ell)$ براساس جمع پیمانه‌ای به پیمانه‌ی n به صورت زیر تعریف می‌شود:

$$(c_1, c_2, \dots, c_\ell) = (m_1 + k_1, m_2 + k_2, \dots, m_\ell + k_\ell) \bmod n$$

- به صورت مشابه عملیات رمزگشایی براساس جمع پیمانه‌ای به پیمانه‌ی n به صورت زیر تعریف می‌شود:

$$(m_1, m_2, \dots, m_\ell) = (c_1 - k_1, c_2 - k_2, \dots, c_\ell - k_\ell) \bmod n$$

حمله به Vigenère Cipher

- بعد از هر ℓ حرف کلید تکرار می‌شود.
- بنابراین مجموعه‌ی حروف رمزشده‌ی $\{c_i, c_{i+\ell}, c_{i+2\ell}, c_{i+3\ell}, \dots\}$، به لحاظ فرکانس تکرار حروف مشابه متن اصلی است.
- برای حمله، ابتدا تلاش می‌کنیم که طول کلید (یعنی ℓ) را به دست آوریم.
- سپس متن رمزشده را به ℓ قسمت تقسیم می‌کنیم به‌گونه‌ای که مجموعه‌ی iام شامل حروفی باشد که با k_i رمز شده‌اند.
- حمله‌ی فرکانسی را بر روی مجموعه‌ها به صورت مجزی اجرا می‌کنیم.
- اما چگونه می‌توان طول کلید را پیدا کرد؟

تست Kasiski

- Kasiski در سال 1863 روشی را به منظور پیدا کردن طول کلید الگوریتم رمزنگاری Vigenère پیشنهاد کرد.
- این روش پیش ازاین توسط Babbage کشف شده، اما مخفی مانده بود.
- مشاهده‌ی اصلی: اگر دو متن برابر با یک کلید ثابت رمز شوند، متون رمز شده‌ی معادل آن‌ها با یکدیگر برابر می شوند.
- فاصله‌ی بین متون رمز شده‌ی برابر، برابر با ضریبی از طول کلید است.
- از این حقیقت می‌توان برای پیدا کردن طول کلید استفاده کرد.

تست Kasiski

مثال

- موقعیت پیام‌های رمزشده‌ی **CHR** در متن رمز شده‌ی روبه‌رو:

1, 166, 236, 276, 286

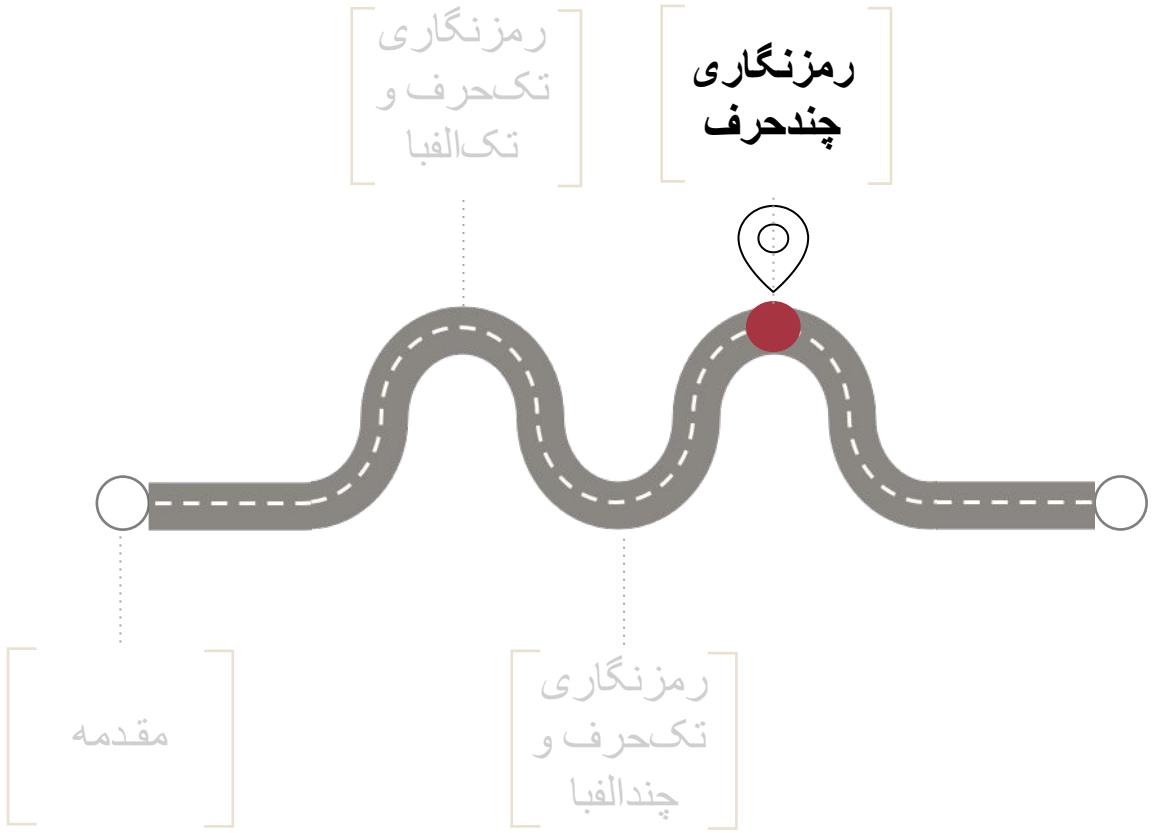
- اختلاف بین این موقعیت‌ها را محاسبه می‌کنیم:

166 – 1 = 165, 236–1=235, 276 – 1 = 275, 286 – 1 = 285

- می‌دانیم که طول کلید تمامی این مقادیر را می‌شمارد.
- با محاسبه بزرگترین مقسوم‌علیه مشترک، می‌توان طول کلید را پیدا کرد.
- برای این مثال طول کلید $m = 5$ محاسبه می‌شود.

CHREEVOAHMAERATBIAXXWTNXBEEOPHBS
BQMQEQERBWRVXUOAKXAOSXXWEAHBW
GJMMQMNKGRFVGXWTRZXWIAKLXFPSKAU
TEMNDCMGTSXMXBTUIADNGMGPSRELXNJE
LXVRVPRTULHDNQWTWDTYGBPHXTFALJHA
SVBFXNGLL**CHR**ZBWELEKMSJIKNBHWRJGN
MGJSGLXFEYPHAGNRBIEQJTAMRVLCRREMN
DGLXRRIMGNSNRW**CHR**QHAEYEVTAQEBBIP
EEWEVKAKOEWADREMXMTBHH**CHR**TKDNV
RZ**CHR**CLQOHPWQAIIWXNRMGWOIIFKEE

Source: “Cryptography: Theory and Practice” by D. Stinson and M. Paterson


رمزنگاری
تک‌حرف و
تک‌الفبا
رمزنگاری
چندحرف
مقدمه
رمزنگاری
تک‌حرف و
چندالفبا

جایگزینی بر مبنای دو حرف

- با توجه به حمله‌ی فرکانسی به نظر می‌رسد که هیچ رمزی که مبتنی بر جایگزینی تک حرف است، نمی‌تواند امن باشد.
- راهکار دیگر استفاده از جایگزینی دو حرف با دو حرف است.
- چالشی که این روش دارد این است که تعداد حالت‌هایی که باید برای کلید ذخیره شوند (طول کلید) بسیار زیاد است: $26 \times 26 = 676$.
- بنابراین به یک روش موثر برای رمزنگاری نیاز است.

جایگزینی دوحرفی در Playfair Cipher

- کلید: حروف انگلیسی که به صورت تصادفی در سلول‌های یک جدول 5 در ۵ قرار دارند.
- برای رمزگذاری، متن را به دوحرفی‌ها تقسیم کرده و به صورت زیر جایگزین می‌کنیم:

1. اگر دو حرف موردنظر در یک سطر بودند: هر حرف را با حرف سمت راست آن جایگزین می‌کنیم.
2. اگر دو حرف موردنظر در یک ستون بودند، هر حرف را با حرف پایین آن جایگزین می‌کنیم.
3. در غیر این صورت، حرف اول را با حرفی که در سطر حرف اول است و حرف دوم را با حرفی که در سطر حرف دوم است، جایگزین می‌کنیم.

S	T	A	N	D
E	R	C	H	B
K	F	G	I\J	L
M	O	P	Q	U
V	W	X	Y	Z

GLOW WORM

GL OW WO RM

IK WT TW EO

Example's source: "Introduction to Cryptography and Security Mechanisms" by Keith Martin

امنیت Playfair Cipher

- می‌توان فرکانس دو حرفی‌ها در زبان مورد هدف را نیز در نظر گرفت.
- به صورت مشابه، هرچند با تعداد متن بیشتری، می‌توان حمله‌ی فرکانسی را بر روی Playfair Cipher اجرا کرد!

سی دوحرفی‌ای که بیشترین تکرار را در زبان انگلیسی دارند (به ترتیب):

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.

Source: “Cryptography: Theory and Practice” by D. Stinson and M. Paterson

Hill Cipher

- کلید K یک ماتریس $\ell \times \ell$ مخفی است که در پیمانه‌ی 26 (تعداد حروف آن زبان) معکوس‌پذیر است.
- متن اصلی به دسته‌های ℓ حرفی تقسیم می‌شود.
- عملیات رمزگذاری ℓ حرف $(m_1, \dots, m_\ell)$ به صورت زیر تعریف می‌شود:

$$(m_1, \dots, m_\ell).K \bmod 26$$

- عملیات رمزگشایی با استفاده از معکوس ماتریس کلید می‌باشد:

$$(c_1, \dots, c_\ell).K^{-1} \bmod 26$$

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1\ell} \\ k_{21} & k_{22} & & k_{2\ell} \\ \vdots & & \ddots & \vdots \\ k_{\ell 1} & k_{\ell 2} & \cdots & k_{\ell\ell} \end{bmatrix}$$

$$\begin{aligned} c_1 &= m_1 k_{11} + m_2 k_{21} + \cdots + m_\ell k_{\ell 1} \\ c_2 &= m_1 k_{12} + m_2 k_{22} + \cdots + m_\ell k_{\ell 2} \\ &\dots \\ c_\ell &= m_1 k_{1\ell} + m_2 k_{2\ell} + \cdots + m_\ell k_{\ell\ell} \end{aligned}$$

امنیت Hill Cipher

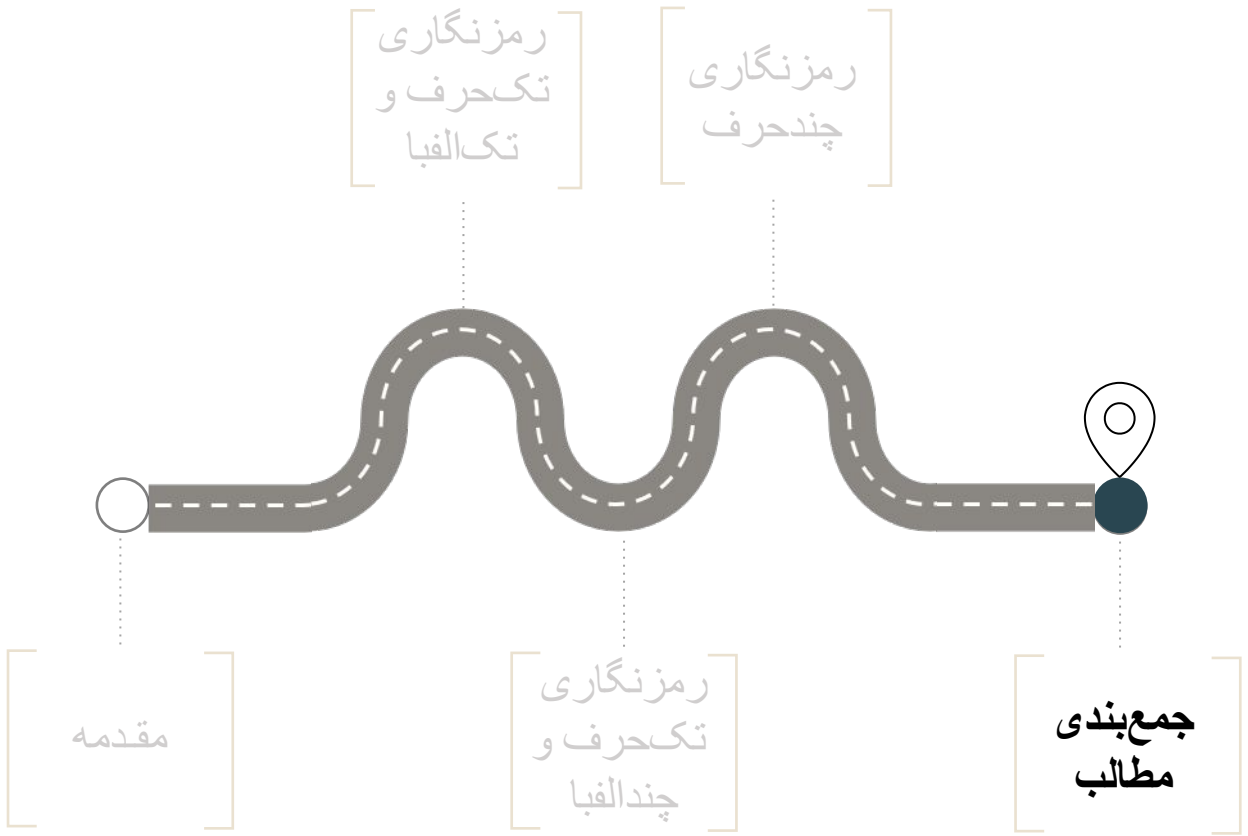
- روش‌های قبلی به راحتی به Hill Cipher قابل اعمال نیستند.
- اما این الگوریتم رمزنگاری در سناریو حمله‌ی متن معلوم امن نیست.
- مثال: فرض کنید که می‌دانیم که $\ell = 2$ است و متن اصلی Friday تبدیل به متن رمزشده PQCFKU شده است. به عبارتی داریم:

$$E_K(5,17) = (15,16), \quad E_K(8,3) = (2,5), \quad E_K(0,24) = (10,20)$$

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K \Rightarrow K = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix}$$

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

Example's source: "Cryptography and Network Security" by William Stallings


رمزنگاری تک‌حرف و تک‌الفبا
رمزنگاری چندحرف
مقدمه
رمزنگاری تک‌حرف و چندالفبا
جمع‌بندی مطالب

جمع‌بندی مطالب

- بین متن اصلی و متن رمزشده‌ی سیستم‌های رمزنگاری کلاسیک روابط آماری قوی‌ای وجود دارد.
- به همین دلیل، این سیستم‌ها عموما در سناریوی متن معلوم به راحتی شکسته می‌شوند.
- این سیستم‌ها امروزه کاربرد عملی چندانی نداشته و تنها جنبه‌ی آموزشی و آشنایی با خط تدریجی رشد علم رمزنگاری را پیدا کرده کرده‌اند.