Shahid Beheshti University

رمزنگاری پیشرفته

هادی سلیمانی

پژوهشکده فضای مجازی دانشگاه شهید بهشتی

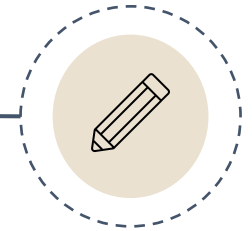
مجوز استفاده و نشر

- 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/advanced-cryptography-course/

درس سوم

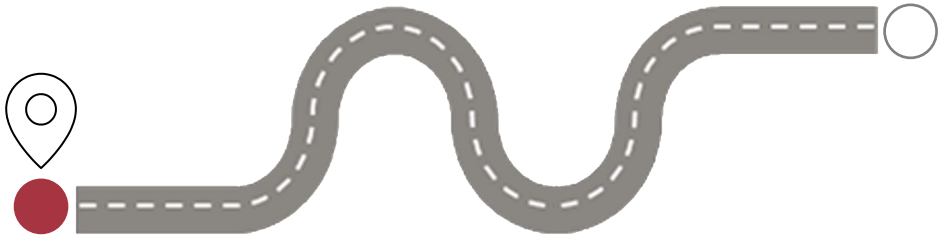


حمله‌ی همبستگی

فهرست عناوین درس

حمله‌ی همبستگی

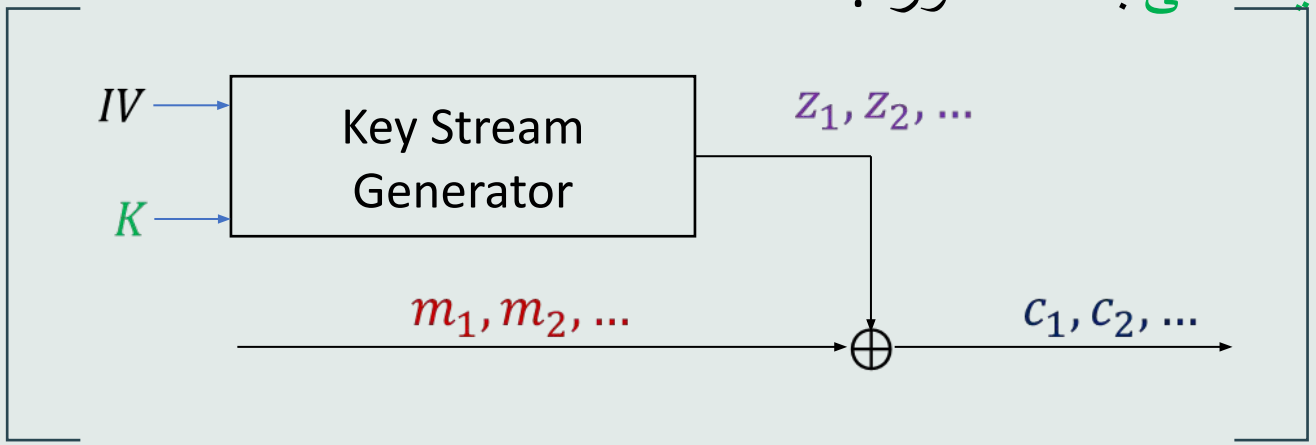
- مقدمه
- حمله‌ی همبستگی پایه
- حمله‌ی همبستگی سریع
- جمع‌بندی



مقدمه

یادآوری: رمز جریانی

- رمز جریانی یک اولیه‌ی رمزنگاری متقارن است که با استفاده از کلید و یک مقدار اولیه (IV)، دنباله‌ای از بیت‌های شبه تصادفی تولید می‌کند. این دنباله را دنباله‌ی کلید اجرایی (Key Stream) می‌نامند.
- دنباله‌ی کلید اجرایی به‌نحوی با متن اصلی ترکیب شده (معمولا xor بیتی) و متن رمز شده را تولید می‌کند.
- ویژگی یک رمز جریانی امن این است که با فرض دانستن بخشی از دنباله‌ی کلید اجرایی (سناریوی متن معلوم)، نتوان با روشی سریع‌تر از جست‌وجوی کامل اطلاعاتی درباره‌ی کلید مخفی به دست آورد.



یادآوری: کاربرد LFSRها در رمزهای جریانی

- LFSRها ویژگی‌های بسیار مناسبی دارند که ظاهرا الزامات اولیه برای استفاده در ساخت رمزهای جریانی را برآورده می‌سازند.

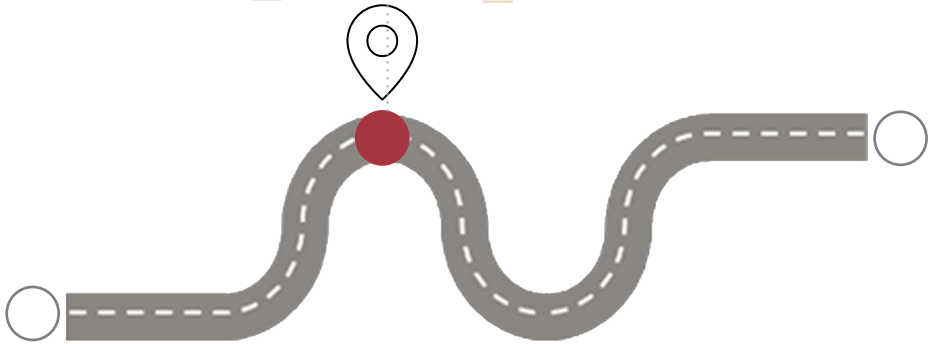
مزایا	• دوره‌ی تناوب زیاد (تقریبا حداکثری) • مشخصات آماری مناسب
معایب	• چالش خطی بودن

سوال: چگونه می‌توان بر چالش خطی بودن LFSRها غلبه کرد، ولی در عین حال از مزایای آن‌ها نیز بهره برد؟

یادآوری: روش‌های متداول برای افزایش پیچیدگی خطی LFSRها

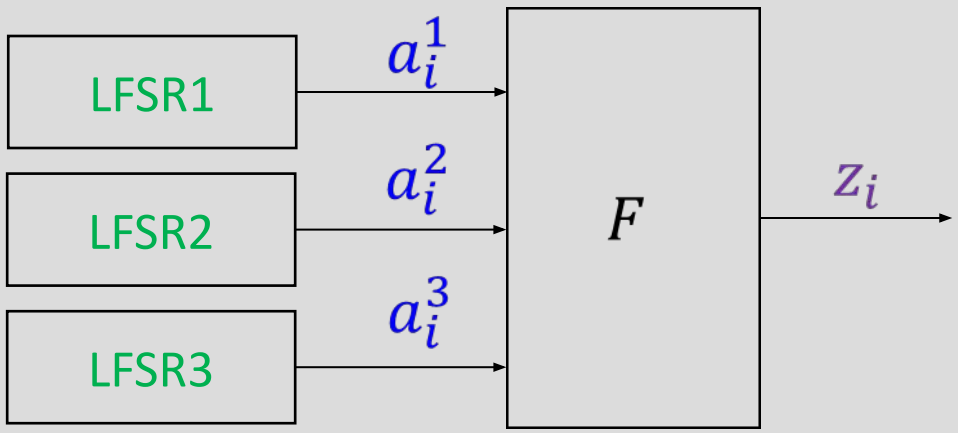
- استفاده از یک مولد فیلتر غیرخطی (Nonlinear Filter Generator)
- ترکیب غیرخطی خروجی‌های چند رجیستر و یا LFSR
- استفاده از پالس‌های نامنظم
  - گام‌های متناوب (Alternating Steps)
  - کاهش غیرمنظم خروجی (Shrinking)
- ...
- ترکیبی از روش‌های فوق

حمله‌ی همبستگی پایه



مقدمه

بررسی امنیت ترکیب غیرخطی خروجی چند LFSR



- فرض کنید که حالت اولیه رجیسترها، همان کلید مخفی باشد.
- در هر کلاک، خروجی رجیسترها تولید و سپس با استفاده از تابع F ترکیب می‌شود.
- خروجی تابع غیرخطی F دنباله‌ی کلید اجرایی است.
- برای یافتن کلید از طریق جست‌وجوی کامل، باید به‌ازای هر مقدار اولیه‌ی ممکن برای رجیسترها، دنباله‌ی خروجی را تولید کنیم و ببینیم آیا با دنباله‌ی z برابر می‌شود یا خیر؟
- اگر طول رجیسترها به ترتیب L_1، L_2 و L_3 باشد، پیچیدگی جست‌وجوی کامل برابر است با: $2^{L_1+L_2+L_3}$.

مثال: استفاده از Geffe Generator

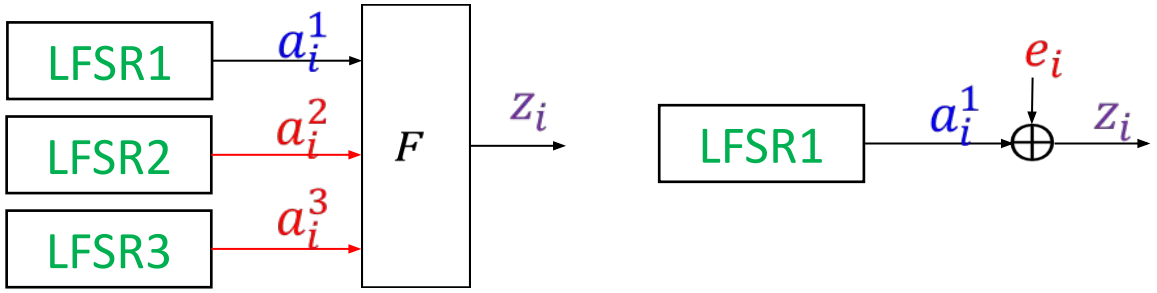
- بین تعداد 0 و 1های خروجی تابع توازن برقرار است (دنباله به ظاهر تصادفی است).
- اما یک همبستگی بالا بین مقدار خروجی تابع (z_i) و مقدار a_i^1 مشاهده می‌شود:

$$\Pr[z_i = a_i^1] = 6/8 = 3/4 \gg 1/2$$

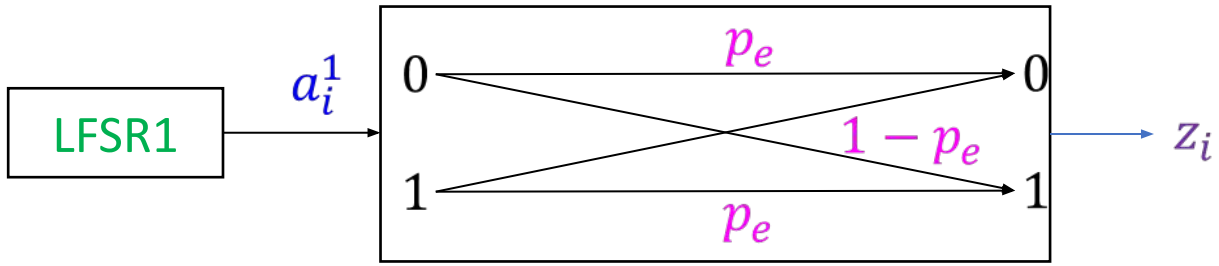
$$z_i = F(a_i^3, a_i^2, a_i^1) = (a_i^1 \wedge a_i^2) \oplus (\neg a_i^3 \wedge a_i^1)$$

0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

مدلی برای حمله‌ی همبستگی



- بیت z_i از دنباله‌ی کلید اجرایی را می‌توان ماحصل اضافه شدن نویز e_i به بیت خروجی LFSR (a_i^1) در نظر گرفت که با احتمال $1-p_e$ (احتمال p_e) بیت را تغییر می‌دهد (نمی‌دهد).



- اگر احتمال $p_e = \Pr[e_i = 0]$ از 1/2 (یعنی احتمال حالت تصادفی) فاصله‌ی قابل توجهی داشته باشد، به معنای آن است که بین دنباله‌های $\boldsymbol{a^1}$ و $\boldsymbol{z}$ یک همبستگی وجود دارد.

حمله‌ی همبستگی پایه

(روش Siegenthaler)

فرض کنیم:

1. خروجی یکی از LFSRها با دنباله‌ی کلید اجرایی همبستگی زیاد دارد (به عبارتی: $|p_e - 1/2| \gg 0$).
2. N بیت از دنباله‌ی کلید اجرایی به دست آمده است. مثلا: $z_0, \dots, z_{N-1}$.

روش حمله:

- حالت اولیه‌ی یک LFSR را حدس زده و N بیت از دنباله‌ی خروجی LFSR را تولید می‌کنیم $(a'_0, \dots, a'_{N-1})$.
- فاصله‌ی همینگ دو دنباله‌ی $\boldsymbol{z}$ و $\boldsymbol{a'}$ را محاسبه می‌کنیم:

$$D = \sum_{i=0}^{N-1} (a'_i \oplus z_i)$$

- اگر D تقریبا برابر با $N.p_e$ بود، حالت اولیه‌ی حدس زده شده صحیح است.

حمله‌ی همبستگی پایه

(روش Siegenthaler)

فرض کنیم:

1. خروجی یکی از LFSRها با دنباله‌ی کلید اجرایی همبستگی زیاد دارد (به عبارتی: $|p_e - 1/2| \gg 0$).
2. N بیت از دنباله‌ی کلید اجرایی به دست آمده است. مثلا: $z_0, \dots, z_{N-1}$.

روش حمله:

- حالت اولیه‌ی یک LFSR را حدس زده و N بیت از دنباله‌ی خروجی LFSR را تولید می‌کنیم $(a'_0, \dots, a'_{N-1})$.
- فاصله‌ی همینگ دو دنباله‌ی $\boldsymbol{z}$ و $\boldsymbol{a'}$ را محاسبه می‌کنیم:

$$D = \sum_{i=0}^{N-1} (a'_i \oplus z_i)$$

- اگر D تقریبا برابر با $N.p_e$ بود، حالت اولیه‌ی حدس زده شده صحیح است.

پیچیدگی حمله

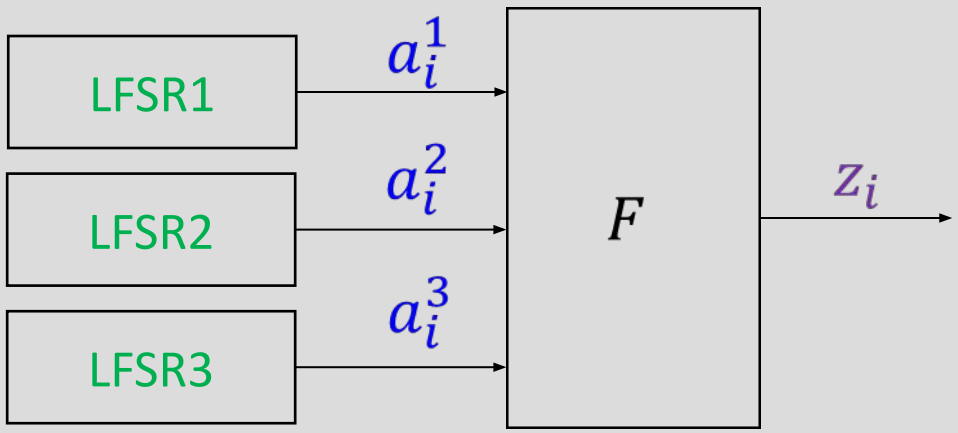
- پیچیدگی داده: اگر طول LFSR برابر با L باشد، تعداد بیت‌های مورد نیاز از کلید اجرایی برای اعمال حمله برابر است با:

$$\mathrm{N} \cong \frac{L}{\left(p_e - \frac{1}{2}\right)^2}$$

- پیچیدگی زمانی: در صورت به‌دست آوردن حالت اولیه‌ی صحیح LFSR، معادل است با محاسبه‌ی N بیت خروجی به‌ازای تمام 2^L کاندید ممکن برای حالت اولیه:

$$2^L N \cong \frac{L.2^L}{\left(p_e - \frac{1}{2}\right)^2}$$

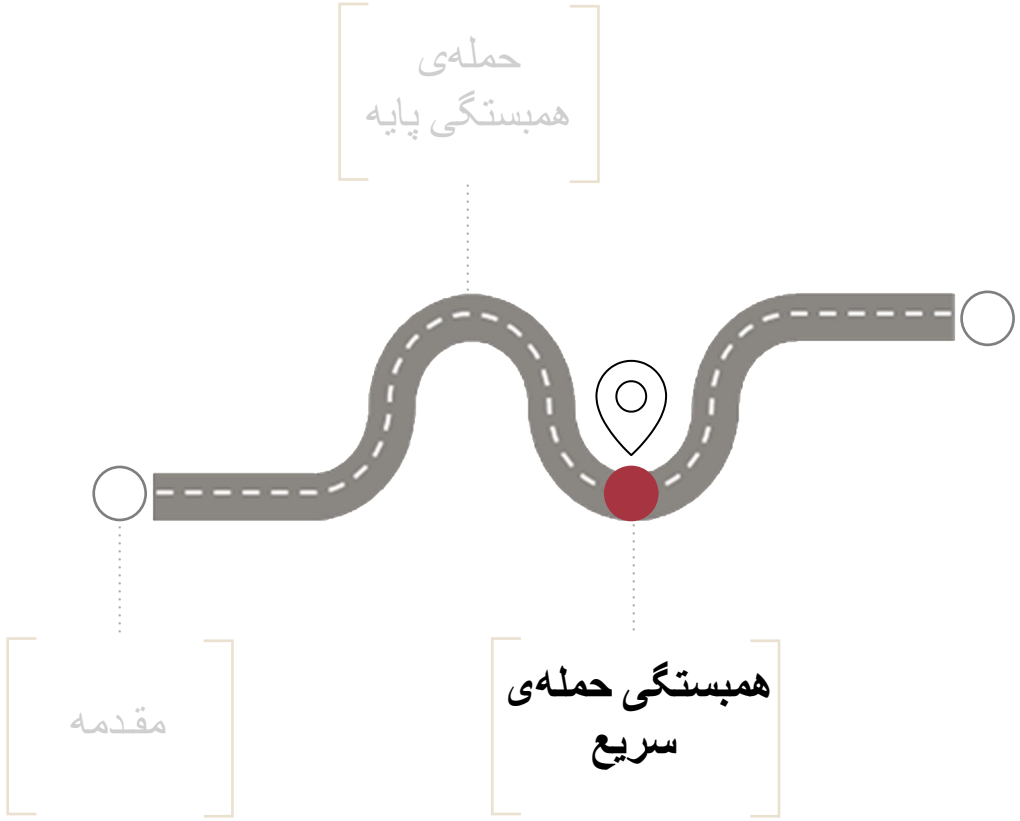
جمع‌بندی بخش



- اگر بین خروجی و ورودی‌های تابع F یک همبستگی بالا وجود داشته باشد، امنیت روش ترکیب غیرخطی خروجی‌ها می‌تواند با استفاده از حمله‌ی همبستگی کاهش پیدا کند.
- پیچیدگی زمانی این حمله تقریبا برابر است با:

$$\frac{L_1 2^{L_1}}{\left(p_{e_1}-\frac{1}{2}\right)^2}+\frac{L_2 2^{L_2}}{\left(p_{e_2}-\frac{1}{2}\right)^2}+\frac{L_3 2^{L_3}}{\left(p_{e_3}-\frac{1}{2}\right)^2}$$

- بسته به مقادیر احتمال‌ها، این مقدار می‌تواند نسبت به پیچیدگی جست‌وجوی کامل $(2^{L_1+L_2+L_3})$ به‌مراتب کمتر باشد.
- اگر تنها حالت اولیه‌ی تعدادی از رجیسترها قابل بازیابی باشد، بقیه حالت‌های اولیه را می‌توان با استفاده از جست‌وجوی کامل به دست آورد.


حمله‌ی
همبستگی پایه
مقدمه
همبستگی حمله‌ی
سریع

حمله‌ی همبستگی سریع در یک نگاه

- ایده‌ی اصلی: استفاده از روابط خطی موجود بین بیت‌های خروجی یک LFSR با a_n.
- مفروضات حمله شبیه به حمله‌ی همبستگی پایه است:
  1. وجود یک همبستگی قوی بین خروجی LFSR (دنباله $\boldsymbol{a}$) و خروجی الگوریتم (دنباله‌ی $\boldsymbol{z}$).
  2. تعداد مورد نیاز از بیت‌های دنباله‌ی کلید اجرایی به دست آمده است.
- مزیت نسبت به حمله‌ی همبستگی پایه:
  - کاهش پیچیدگی زمانی.
- محدودیت‌های حمله‌ی همبستگی سریع نسبت به روش پایه:
  - زمانی کارا است که تعداد ضریب‌های مخالف صفر چندجمله‌ای فیدبک کم باشد.
  - به تعداد متن‌های بیشتری نیاز دارد، یعنی مهاجم به تعداد بیت های بیشتری از دنباله‌ی کلید اجرایی نیاز دارد.

تعریف معادلات خطی برای بیت a_n

- چندجمله‌ای فیدبک LFSR را به صورت زیر در نظر می‌گیریم و فرض می‌کنیم که t مقدار از مجموعه‌ی ضریب‌های آن مقدار 1 دارند:

$$c(x) = c_0 + c_1 x + \cdots + c_k x^k$$

- از طرفی، رابطه‌ی خطی زیر همواره برقرار است:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

- با شیفت دادن اندیس a_n در معادله، مشاهده می‌شود که می‌توان برای بیت a_n، t معادله‌ی خطی برحسب سایر بیت‌های $\boldsymbol{a}$ تعریف کرد.
- مثال: برای یک LFSR با چند جمله‌ای فیدبک $c(x) = x^3 + x + 1$:

$$a_{n-3} + a_{n-1} + a_n = 0$$

$$a_{n-2} + a_n + a_{n+1} = 0$$

$$a_n + a_{n+2} + a_{n+3} = 0$$

تعریف معادلات خطی بیشتر برای بیت a_n

- همان‌طور که براساس چندجمله‌ای $c(x)$ برای بیت a_n، می‌توان t رابطه‌ی خطی تعریف کرد، به طور مشابه می‌توان براساس چندجمله‌ای $c(x)^j$ نیز t رابطه‌ی خطی دیگر برای بیت a_n تعریف کرد.
- یکی از روش‌های به‌دست آوردن روابط خطی بیشتر $(c(x)^j)$، مربع کردن متوالی چندجمله‌ای $c(x)$ است، یعنی: $j = 2^i$
- مثال: برای یک LFSR با تابع فیدبک $c(x) = x^3 + x + 1$ داریم:

$$c(x)^2 = x^6 + x^2 + 1$$

$$a_{n-6} + a_{n-2} + a_n = 0$$

$$a_{n-4} + a_n + a_{n+2} = 0$$

$$a_n + a_{n+4} + a_{n+6} = 0$$

- مزیت مربع‌رسانی این است که تعداد ضریب‌های غیرصفر (t) را ثابت نگه می‌دارد.

نمایش معادلات خطی به‌دست‌آمده

- با روش بیان‌شده، m معادله‌ی خطی که شامل a_n هستند، تشکیل می‌دهیم و آن‌ها را به شکل زیر در نظر می‌گیریم:

$$\ell_1 = a_n + b_1 = 0$$
$$\vdots$$
$$\ell_m = a_n + b_m = 0$$

که b_i مجموع t بیت از دنباله $\boldsymbol{a}$ است:

$$b_i = a_{i_1} + a_{i_2} + \cdots + a_{i_t}$$

- مهاجم به مقادیر دنباله‌ی $\boldsymbol{a}$ دسترسی ندارد، اما می‌داند که بین بیت‌های دنباله‌ی کلید اجرایی و دنباله‌ی خروجی LFSR یک همبستگی با احتمال p وجود دارد.

جای‌گذاری در معادله

- در معادلات خطی، بیت‌های دنباله‌ی $\boldsymbol{a}$ را با بیت‌های متناظر دنباله‌ی کلید اجراسی ($\boldsymbol{z}$) جایگزین کرده و روابط خطی را به شکل زیر در نظر می‌گیریم:

$$\ell_1 = z_n + y_1$$
$$\vdots$$
$$\ell_m = z_n + y_m$$

که y_i مجموع t بیت از دنباله $\boldsymbol{z}$ است:

$$y_i = z_{i_1} + z_{i_2} + \cdots + z_{i_t}$$

- هدف: با در نظر گرفتن مقادیر $\ell_1, \ldots, \ell_m$ که براساس مقادیر دنباله‌ی کلید اجرایی محاسبه شده‌اند، (با احتمال بالاتری) قضاوت کنیم که آیا $a_n = z_n$ هست یا خیر؟
- به‌طور دقیق‌تر، می‌خواهیم بررسی کنیم که: آیا تعداد دفعاتی که مقادیر تصادفی $\ell_1, \ldots, \ell_m$ صفر می‌شوند، با برابر بودن یا نبودن رابطه‌ی $a_n = z_n$،

مشاهده: وجود یک تمایزگر

- احتمال اینکه h تا از مقادیر $\ell_1, \ldots, \ell_m$ صفر شوند:

$$\Pr[h \text{ equations hold}]$$

$$= \begin{cases} \Pr[y_i = b_i]^h \Pr[y_i \neq b_i]^{m-h} & \text{if } z_n = a_n \\ \Pr[y_i \neq b_i]^h \Pr[y_i = b_i]^{m-h} & \text{if } z_n \neq a_n \end{cases}$$

- احتمال اینکه h تا از مقادیر $\ell_1, \ldots, \ell_m$ صفر شوند، براساس این‌که مقدار بیت z_n (که برای مهاجم معلوم است) با بیت a_n (که مهاجم می‌خواهد به دست بیاورد) برابر است یا خیر، متفاوت است.
- مفهوم: براساس جایگذاری کلید اجرایی در معادلات خطی، ممکن است بتوان اطلاعاتی را در خصوص این‌که $a_n = z_n$ یا خیر، استنتاج کرد.
- اما چگونه می‌توان مقدار $\Pr[y_i = b_i]$ را محاسبه کرد؟

احتمال برابر بودن y_i و b_i

- احتمال $S = \Pr(y_i = b_i)$ را می‌توان به صورت بازگشتی محاسبه کرد:

$$S(p,1) = p$$
$$S = S(p,t) = p.S(p,t-1) + (1-p).(1-S(p,t-1))$$

- حال می‌توان احتمال این‌که h تا از مقادیر $\ell_1, \ldots, \ell_m$ صفر شوند را به صورت دقیق محاسبه کرد:

$$\Pr[h \text{ equations hold}]$$
$$= \begin{cases} \Pr[y_i = b_i]^h \Pr[y_i \neq b_i]^{m-h} & \text{if } z_n = a_n \\ \Pr[y_i \neq b_i]^h \Pr[y_i = b_i]^{m-h} & \text{if } z_n \neq a_n \end{cases}$$
$$= \begin{cases} S^h (1-S)^{m-h} & \text{if } z_n = a_n \\ (1-S)^h S^{m-h} & \text{if } z_n \neq a_n \end{cases}$$

جمع‌بندی محاسبه‌ی احتمال

$$p^* = \Pr[z_n = a_n | h \text{ equations hold}] =$$

$$\frac{\Pr[z_n = a_n, h \text{ equations h}}{\Pr[z_n = a_n, h \text{ equations hold}] + \Pr[z_n \neq a}$$

$$\frac{pS^h(1-S)^{m-h}}{pS^h(1-S)^{m-h} + (1-p)(1-S)^{\mathrm{h}}S^{\mathrm{m-h}}}$$

- انتظار داریم برای حالتی که $z_n = a_n$، $p^* > p$ باشد.
- مثال: برای رجیستر با طول 100 بیت، دنباله‌ی کلید اجرایی 5000 هزار بیت، احتمال $p = 0.75$ و $t = 2$:

12	0.9993
11	0.9980
10	0.9944

الگوریتم‌های حمله‌ی همبستگی سریع

- Meier و Staffelbach دو الگوریتم برای استفاده از احتمال p^* معرفی کردند که به الگوریتم‌های A و B مشهور هستند.
- این الگوریتم‌ها تنها در صورتی کارا هستند که تعداد ضرایب غیرصفر LFSR کمتر از 10 باشد $(t < 10)$.

الگوریتم A:

- پیچیدگی زمانی نمایی دارد.
- با در نظر گرفتن بیت‌هایی با بالاترین احتمال p^*، یک حالت اولیه به‌دست می‌آورد. سپس تلاش می‌کند که حالت اولیه‌ی اصلی را بازیابی کند.

الگوریتم B:

- پیچیدگی زمانی چندجمله‌ای دارد.
- همانند الگوریتم A از احتمال p^* استفاده می‌کند.

الگوریتم A

1. پارامترهای حمله (مانند مقدار h) را مشخص می‌کنیم.
   - برای جزییات بیشتر به مرجع انتهای اسلاید مراجعه شود.
2. تعداد k بیت z_i را که دارای بیشترین احتمال p^* هستند انتخاب می‌کنیم (k طول LFSR است) و فرض می‌کنیم که به‌ازای آن‌ها، a_i ها برابر با مقادیر انتخابی z_i هستند. این حالت را حالت اولیه (I_0) در نظر می‌گیریم.
3. تلاش می‌کنیم مقدار صحیح را با تغییرات کوچک در I_0 به‌دست آوریم (با تغییر یکی از بیت‌ها، دو تا از بیت‌ها، سه تا از بیت‌ها و همین‌طور به‌ترتیب، تا جایی که به جواب برسیم).

پیچیدگی حمله

- اگر فاصله‌ی همینگ حالت صحیح از حالت اولیه، r باشد، پیچیدگی حمله برابر $\sum_{i=0}^{r}\binom{k}{i}$ می‌شود.
- کران بالای پیچیدگی به صورت $\sum_{i=0}^{r}\binom{k}{i} \leq 2^{H(\theta).k}$ مشخص می‌شود که $H(\theta)$ در آن برابر است با:

$$H(\theta) = -\theta \log \theta - (1-\theta)\log(1-\theta)$$

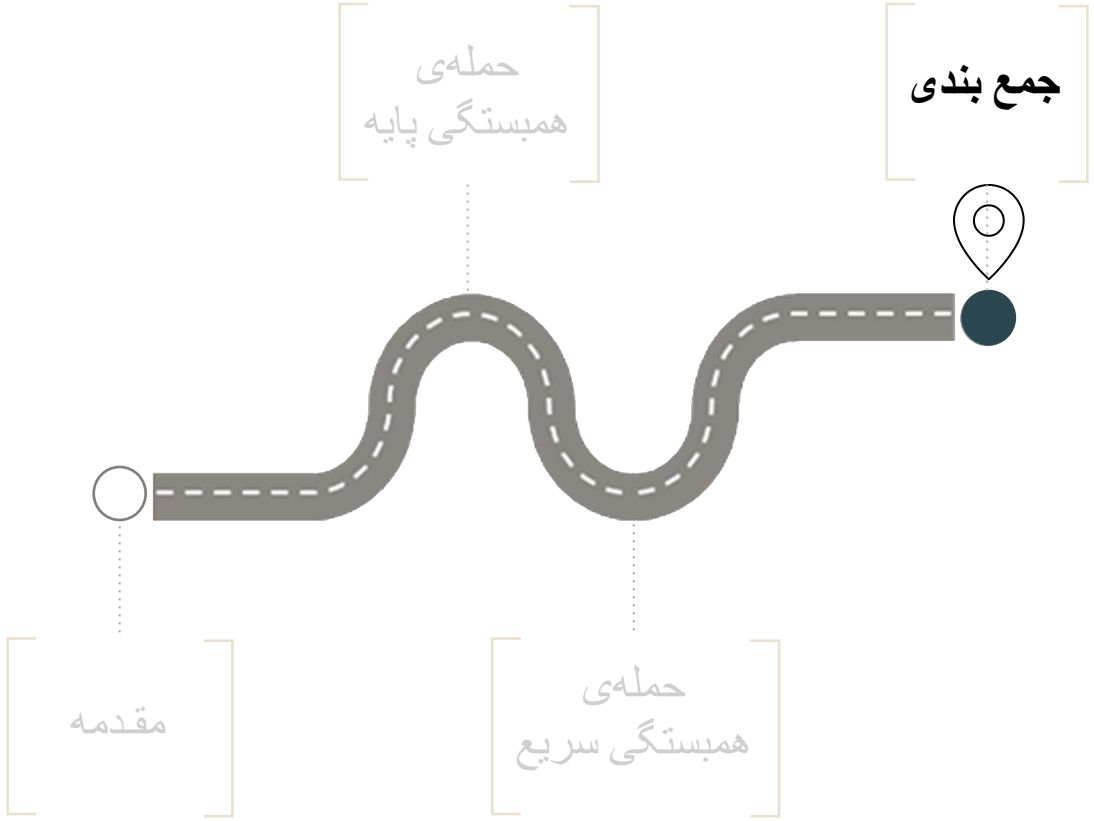
- پیچیدگی حمله از مرتبه‌ی $O(2^{c.k})$ می‌شود که در آن $c < 1$ است (کلید پیش از امتحان کردن تمامی حالات به‌دست می‌آید).
- ضریب c تابعی از t (تعداد ضرایب غیرصفر تابع فیدبک)، احتمال p و میزان بیت‌های دنباله‌ی کلید اجراییِ در دسترس مهاجم نسبت به طول رجیستر (N/L) است.
- مثال: به ازای $t = 2$، $p = 0.75$ و $N/L = 100$، ضریب c برابر 0.012 است.

الگوریتم B

- در الگوریتم B نیز از احتمال $p^* = \Pr[z_n = a_n | h \text{ equations hold}]$ استفاده می‌شود.

1. به هر بیت دنباله‌ی z، یک احتمال همبستگی اختصاص داده می‌شود.
2. به هر بیت دنباله‌ی z، یک احتمال p^* جدید اختصاص داده می‌شود و این گام α بار تکرار می‌شود.
3. آن دسته از بیت‌های دنباله‌ی z که برای آن‌ها $p^* < p_{thr}$ است، مکمل می‌شوند (0 به 1 و 1 به 0 تبدیل می‌شود).
4. اگر دنباله‌ی z دنباله‌ی صحیح بود، اجرای الگوریتم متوقف می‌شود. در غیر این‌صورت گام اول از ابتدا اجرا می‌شود.

- تعداد دفعات تکرار در گام دوم (مقدار α) و هم‌چنین احتمال p_{thr} باید به‌صورت مناسب انتخاب شوند تا اجرای الگوریتم بهینه شود.


جمع بندی
حمله‌ی
همبستگی پایه
حمله‌ی
همبستگی سریع
مقدمه

جمع‌بندی

- دنباله‌ی خروجی تولید شده توسط رمزهای جریانی یک دنباله‌ی شبه‌تصادفی است.
- در مواردی میان این دنباله‌ی تولیدشده و خروجی‌های LFSRهای مولد دنباله، یک همبستگی معنادار وجود دارد.
- می‌توان با استفاده از این همبستگی، کلید مخفی الگوریتم را با پیچیدگی کمتر از جستجوی کامل به‌دست آورد که به آن حمله‌ی همبستگی گفته می‌شود.

معرفی مـراجع تکمیلی جهت مطالعه‌ی بیشتر

حمله‌ی همبستگی

1. T. Siegenthaler. “Decrypting a class of stream ciphers using ciphertext only”. IEEE Trans. Comput., 34:81–85, 1985.
2. W. Meier and O. Staffelbach, “Fast Correlation Attacks on Certain Stream Ciphers”. J. Cryptology, vol 1, number 3, pages 159-176, 1989.