به نام خدا

دانشکده مهندسی برق،
دانشگاه صنعتی شریف

مبانی رمزنگاری و امنیت شبکه

سیستم‌های رمز دنباله‌ای

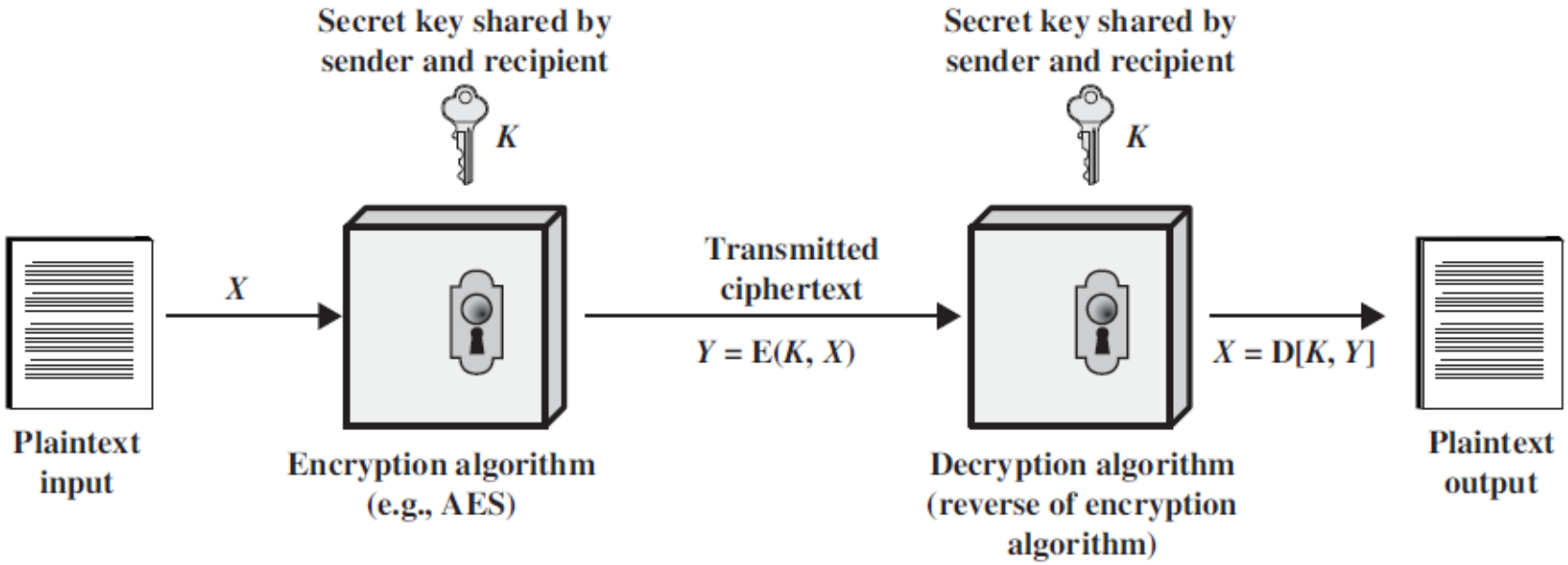
Stream Ciphers

مهتاب میرمحسنی

نیم‌سال دوم (بهار) ۹۸-۹۹

سیستم رمز متقارن یا تک کلیدی (Symmetric=One Key)

- کلیدهای رمزگذاری و رمزگشایی یکسان یا به راحتی از روی یکدیگر قابل محاسبه
- سیستم‌های رمز قالبی (Block Ciphers)
- سیستم‌های رمز دنباله‌ای (Stream Ciphers)



سیستم‌های رمز دنباله‌ای (Stream Ciphers)

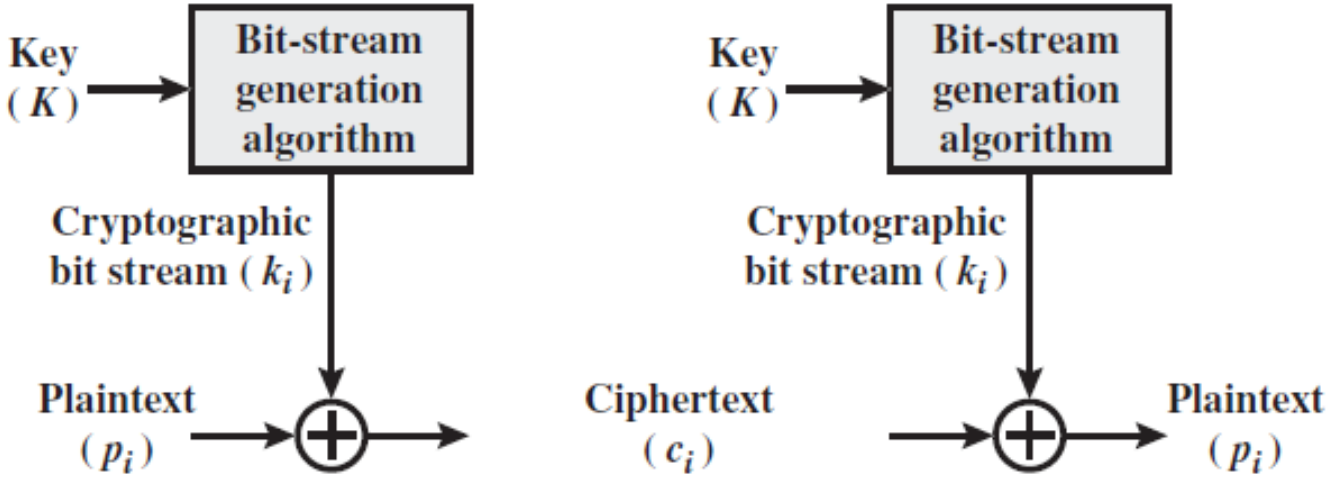
- هر سمبل از دنباله متن اصلی توسط سمبل متناظر دنباله کلید رمز می‌شود
  - معمولاً بیت یا بایت

$$P = P_1 P_2 P_3 \ldots$$

$$K = K_1 K_2 K_3 \ldots$$

$$C = E(K, P) = E(K_1, P_1) E(K_2, P_2) \ldots = C_1 C_2 \ldots \Rightarrow C_i = E(K_i, P_i)$$

- ایده‌آل: دنباله متن اصلی نامحدود ← دنباله کلید نامحدود
  - الگوریتم ساخت دنباله کلید نامتناهی از کلید اصلی

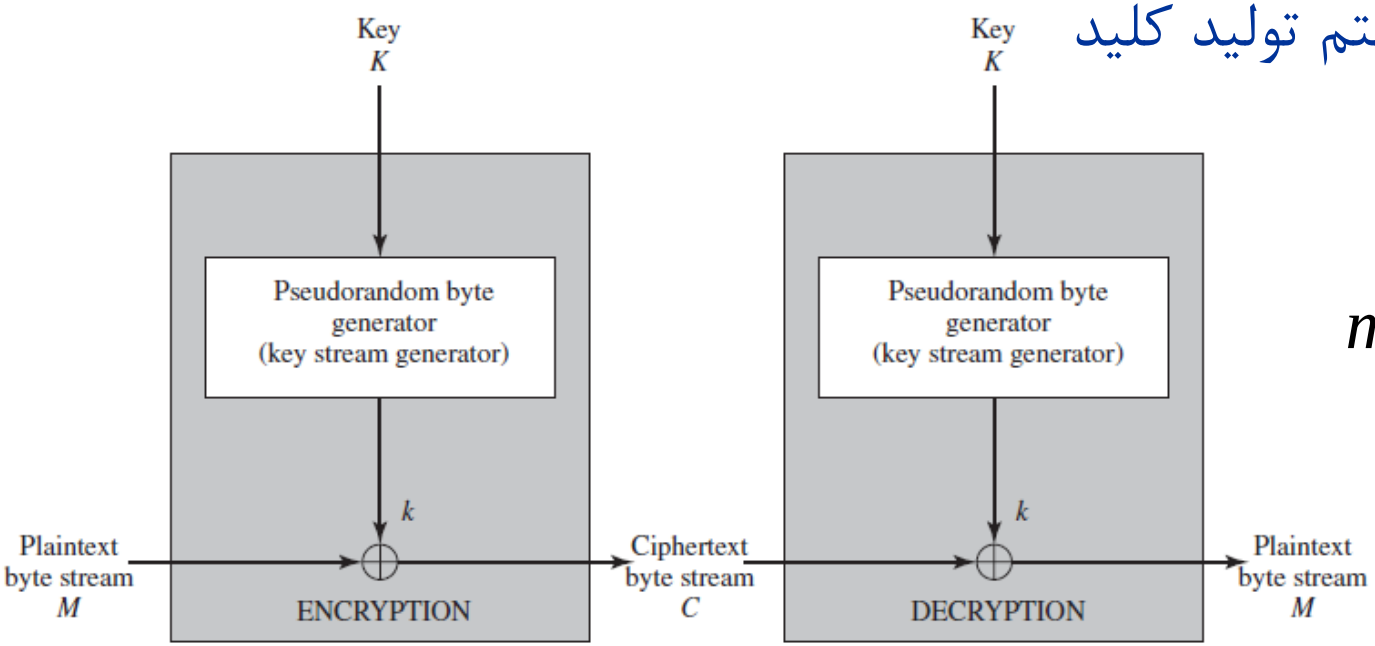


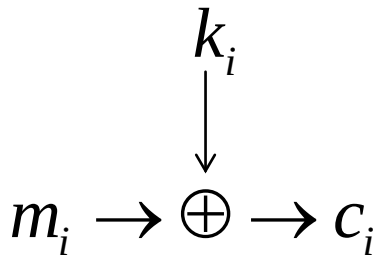
رمز دنباله‌ای

- دنباله کلید
  - نامتناوب: رمز ورنام ← امن کامل
  - متناوب: مانند رمز Vigenère

$$K = K_1 K_2 \ldots K_d K_1 K_2 \ldots K_d \ldots$$

- الگوریتم رمزگذاری ($E(K_i, \cdot)$) ساختار ساده XOR است (طبق معیار شانون)
- پیچیدگی: الگوریتم تولید کلید





رمز دنباله‌ای

- رمزگذاری $C_i = K_i \oplus P_i$

```
  11001100  plaintext
⊕ 01101100  key stream
  10100000  ciphertext
```

- رمزگشایی $P_i = K_i \oplus C_i$

```
  10100000  ciphertext
⊕ 01101100  key stream
  11001100  plaintext
```

$K_i = ?$

الزامات رمز دنباله‌ای

- A1: دوره تناوب دنباله کلید باید از یک حداقل معلوم بزرگ‌تر باشد
  - هر چه طولانی‌تر رمزشکنی پیچیده‌تر (حمله نوع اول)
- A2: دنباله کلید تولید شده تصادفی به نظر برسد
  - خواص دنباله تصادفی واقعی را تا حد ممکن ارضا نماید
  - هر چه دنباله کلید تصادفی‌تر ← متن رمزشده تصادفی‌تر ← رمزشکنی پیچیده‌تر (حمله نوع اول)
- A3: غیرقابل پیش‌بینی: الگوریتم تولید کلید غیرخطی باشد
  - مقابله با حمله نوع دوم (حمله متن اصلی معلوم)
- کلید اصلی طولانی باشد
  - مقابله با حمله جستجوی فراگیر
  - فناوری امروزه: حداقل ۱۲۸ بیت

دنباله‌های تصادفی

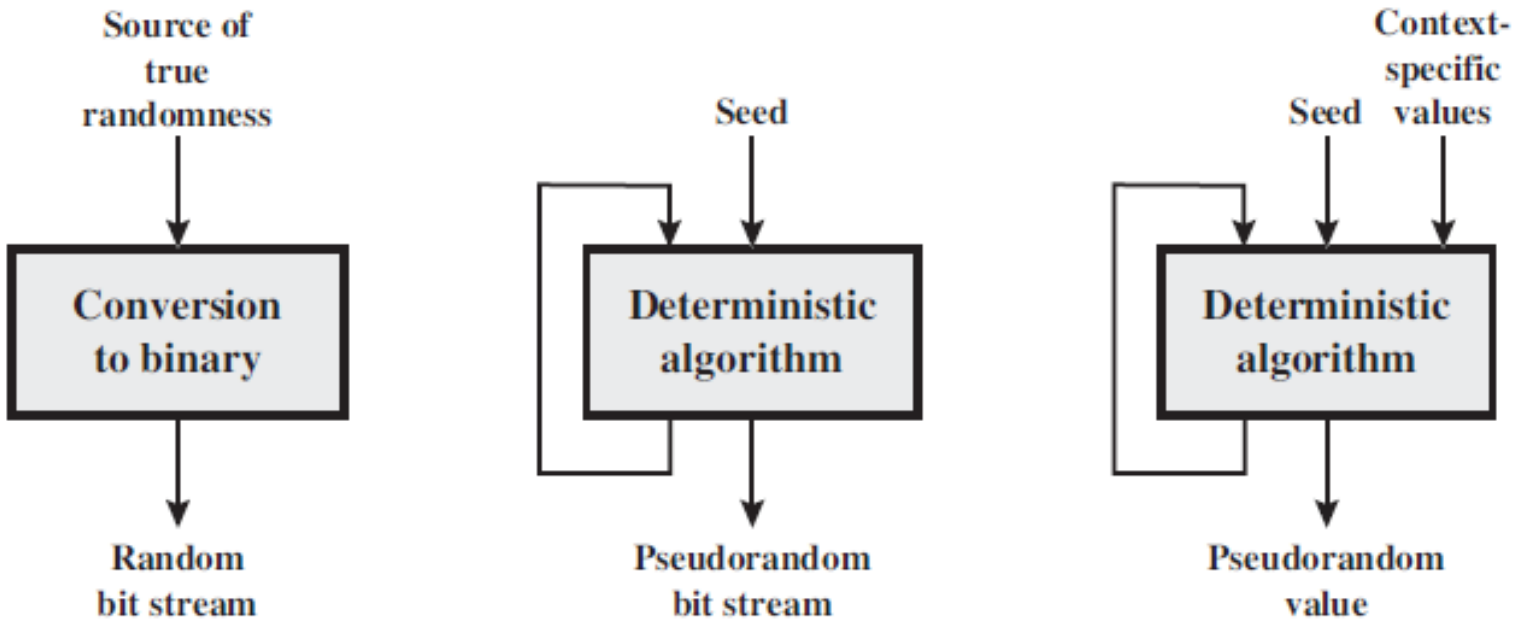
- مولد دنباله شبه‌تصادفی: تولید توسط یک الگوریتم یقینی
  - pseudorandom number generators (PRNGs)
  - deterministic random bit generators (DRBGs)

- استفاده از منبع تصادفی فیزیکی
  - true random number generators (TRNGs)
  - non-deterministic random bit generators (NRBGs)

کاربرد دنباله‌های تصادفی در امنیت

- **دنباله کلید در رمز دنباله‌ای (متقارن)**
- تولید کلید در الگوریتم‌های رمزنگاری کلید همگانی (RSA)
  - تولید اعداد اول (مثلا از مرتبه $۱۰^{۱۵۰}$)
    - روش جستجوی فراگیر (آزمودن تمامی اعداد فرد کوچک‌تر از جذر)
    - استفاده از الگوریتم‌هایی با ورودی اعداد تصادفی- randomization
- تولید کلید نشست (session key)
  - کلید مخفی برای استفاده در مدت زمان معین و محدود
- توزیع کلید و احراز اصالت
  - تولید تک‌شمار (nonce) برای مقابله با حمله تکرار

تولید اعداد تصادفی



TRNG = true random number generator
PRNG = pseudorandom number generator
PRF = pseudorandom function

دنباله شبه‌تصادفی Pseudorandom Number (PN)

- دنباله غیرتصادفی (یقینی) با خواص تصادفی مورد نظر
  - معیارهای سه‌گانه گالوب
  - هر دنباله که در این معیارها صدق کند: PN-sequence
- run: مجموعه سمبل‌های هم‌نام که با سمبل قبل و بعد خود متفاوت باشند
  - gap: اگر سمبل‌ها صفر باشند
  - block: اگر سمبل‌ها یک باشند

دنباله شبه‌تصادفی Pseudorandom Number (PN)

- دنباله غیرتصادفی (یقینی) با خواص تصادفی مورد نظر
  - معیارهای سه‌گانه گالوب
  - هر دنباله که در این معیارها صدق کند: PN-sequence
- run: مجموعه سمبل‌های هم‌نام که با سمبل قبل و بعد خود متفاوت باشند
  - gap: اگر سمبل‌ها صفر باشند
  - block: اگر سمبل‌ها یک باشند
- تابع همبستگی یک دنباله متناوب (با دوره تناوب p): $\{s_t\},\{s_{t+\tau}\}$
  - A: تعداد محل‌هایی که ۲ دنباله در یک دوره تناوب برابرند
  - D: تعداد محل‌هایی که ۲ دنباله در یک دوره تناوب متفاوتند ($D=p-A$)

$$C_\tau = \frac{A-D}{p} \qquad 0 \le \tau < p$$

معیارهای گالوب (تصادفی بودن (A2))

- R1: تعداد صفرها و یک‌ها در یک دوره تناوب برابر باشند
  - p زوج: نصف سمبل‌ها صفر و نصف آن ها یک باشند
  - p فرد: اختلاف تعداد صفرها و یک‌ها، فقط یک باشد
  - توزیع یکنواخت
- R2: یکنواختی runها
  - در یک دوره تناوب، نصف runها دارای طول ۱، یک چهارم دارای طول ۲، یک هشتم دارای طول ۳ و ... باشند
  - برای هر یک از runها، تعداد gapها و blockها برابر باشند
- R3: تابع همبستگی غیرهم‌فاز ($\tau \neq 0$) مقدار ثابتی باشد
  - پس از حذف بایاس: تابع همبستگی غیرهم‌فاز صفر باشد
  - استقلال

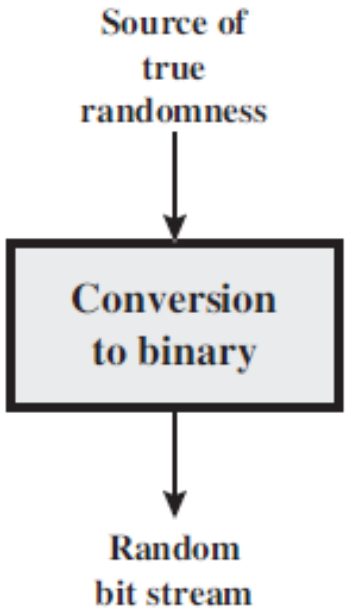
تست‌های آماری ارزیابی تصادفی بودن

- تست فرکانس: بررسی مساوی بودن (تقریبی) احتمال تولید بیت صفر و یک
  - اساسی‌ترین تست
- تست runها: تعداد runها با طول متفاوت تقریبا برابر با مقدار مورد نظر است
- تست سریال: قابل قبول بودن احتمال‌های انتقالی
- تست تابع همبستگی: محاسبه تقریبی
- تست آماری جامع Maurer
  - دنباله کاملاً تصادفی قابل فشرده‌سازی نیست
  - تعیین میزان فشرده‌سازی

عدم پیش‌بینی

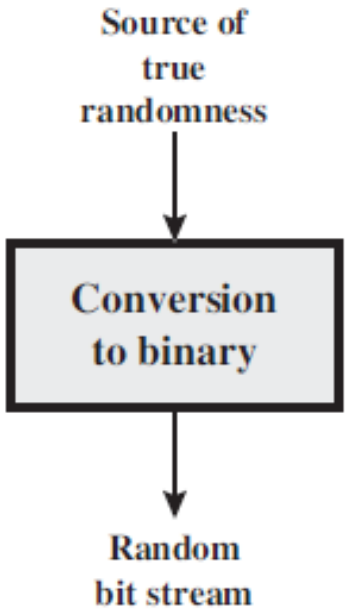
- با دانستن تعدادی از بیت‌های دنباله نتوان دنباله را یافت
- عدم پیش‌بینی پیش‌رو (Forward unpredictability)
  - تا زمانی که کلید اصلی (seed) نامعلوم باشد، با دانستن بیت‌های قبلی دنباله کلید متناوب نتوان بیت بعدی را تعیین کرد
- عدم پیش‌بینی پس‌رو (Backward unpredictability)
  - با دانستن هر تعداد از بیت‌های دنباله کلید متناوب، نتوان کلید اصلی (seed) را تعیین کرد
- تست‌های آماری: مشابه تست‌های ارزیابی تصادفی بودن

true random number generator (TRNG)



- منبع تصادفی (منبع آنتروپی)
  - فیزیکی: حرکت ماوس، مقدار لحظه‌ای کلاک سیستم، ...
  - ترکیب منابع
  - نیاز به حذف بایاس

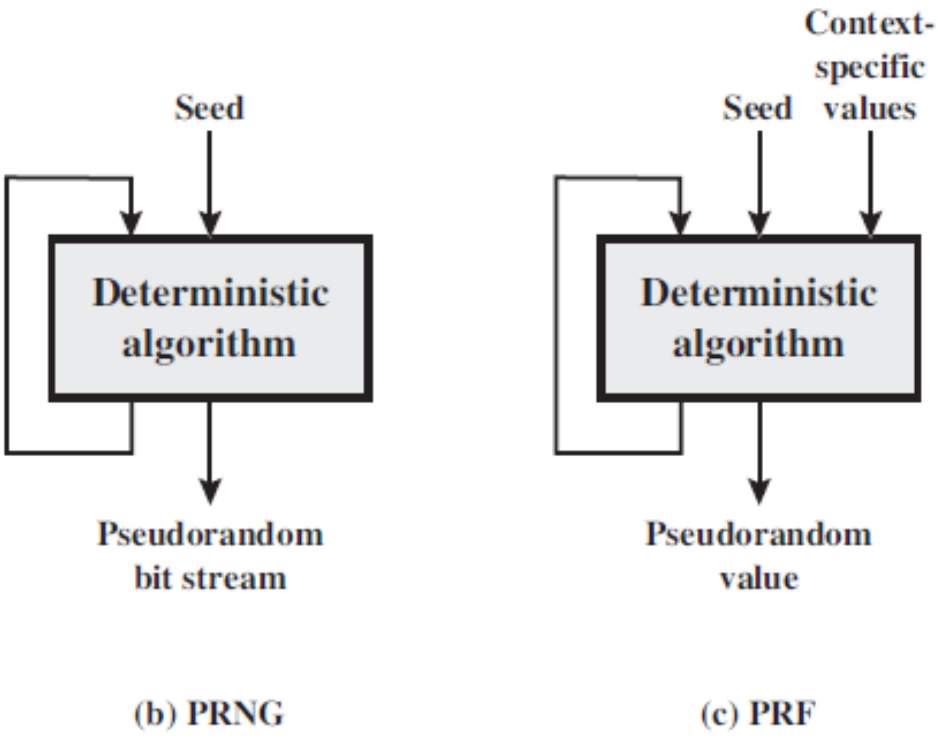
true random number generator (TRNG)



- منبع تصادفی (منبع آنتروپی)
  - فیزیکی: حرکت ماوس، مقدار لحظه‌ای کلاک سیستم، ...
  - ترکیب منابع
  - نیاز به حذف بایاس
- چرا شبه‌تصادفی؟
  - رمز دنباله‌ای: نیاز به ارسال امن دنباله کلید (به طول متن اصلی)
    - ۱۲۸ بیت کلید اصلی
  - کاربردهای دنباله تصادفی با طول محدود ← شبه‌تصادفی
    - حذف بایاس با استفاده از PRF
    - نرخ تولید اعداد واقعی تصادفی معمولا در حد مورد نیاز نیست

شبه تصادفی

- الگوریتم یقینی (deterministic)
- فیدبک از خروجی به ورودی
- مهاجم با دانستن الگوریتم و ورودی (کلید اصلی = seed)، دنباله را می‌تواند بازسازی کند
- PRNG: تولید دنباله با طول دلخواه
  - کاربرد در رمز دنباله‌ای
- PRF: دنباله با طول معین
  - کلید مخفی نشست و تک‌شمار
- الگوریتم یکسان می‌تواند بکار رود
- تست‌های آماری

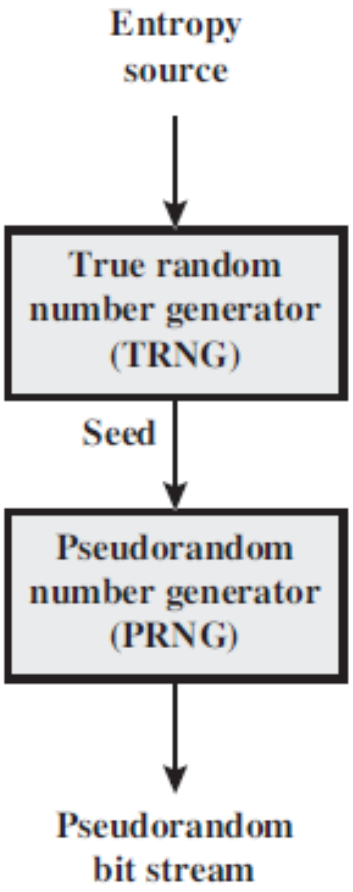


PRNG = pseudorandom number generator

PRF = pseudorandom function

الزامات کلید اصلی = seed

- اگر مهاجم کلید اصلی را بداند با توجه به یقینی بودن الگوریتم، دنباله کلید را می‌سازد
- کلید اصلی باید امن باشد
  - دنباله تصادفی یا شبه تصادفی
- معمولاً کلید اصلی توسط TRNG تولید می‌شود
  - استاندارد SP800-90



طراحی PRNG برای رمزنگاری

- الگوریتم‌های ویژه تولید اعداد شبه تصادفی
  - تنها برای تولید اعداد شبه تصادفی (مولد اعداد شبه تصادفی)
  - برخی استفاده کلی دارند و برخی برای رمز دنباله‌ای طراحی شده‌اند
  - LFSR و RC4
- الگوریتم‌هایی بر پایه روش‌های موجود رمزنگاری
  - الگوریتم‌های رمزنگاری ورودی را تصادفی می‌کنند
    - مهاجم از خواص غیرتصادفی متن رمزشده در رمزشکنی استفاده می‌کند
  - کاربرد: سیستم‌هایی که از الگوریتم خاصی برای رمزنگاری استفاده می‌کنند
    1. رمزهای متقارن قالبی
    2. رمزهای کلید همگانی (نامتقارن): استفاده از مفاهیم نظریه اعداد
    3. توابع چکیده‌ساز و کدهای احراز اصالت پیام

مولد خطی هم‌نهشتی (Linear Congruential Generators)

- پیشنهاد توسط Lehmer در سال ۱۹۵۱
- دنباله اعداد تصادفی $\{X_n\}$ به صورت زیر (تکراری) تولید می‌شود:

$$X_{n+1} = (aX_n + c) \bmod m$$

m	the modulus	$m > 0$
a	the multiplier	$0 < a < m$
c	the increment	$0 \le c < m$
X_0	the starting value, or seed	$0 \le X_0 < m$

- $\{X_n\}$: عدد صحیح $0 \le X_n < m$
- انتخاب مقادیر a, c, m

$$a = 7, c = 0,\ m = 32 \rightarrow \{7, 17, 23, 1, 7, \ldots\}$$

$$a = 5, c = 0,\ m = 32 \rightarrow \{5, 25, 29, 17, 21, 9, 13, 1, 5, \ldots\}$$

$$m \approx 2^{31}$$

مولد خطی هم‌نهشتی (ادامه)

- شرایط پیشنهادی برای مولد:
  - دوره تناوب کامل باشد (تمامی اعداد از 0 تا m-1 تولید شوند)
    - m اول و c=0 باشد ← می‌توان a را طوری انتخاب کرد که دوره تناوب m-1 باشد (همه مقادیر به جز 0)
  - دنباله تولیدشده تصادفی به نظر برسد
  - قابل پیاده‌سازی در حساب ۳۲ بیتی
    - m اول مناسب $m = 2^{31} - 1$
- انتخاب مناسب مقادیر a, c, m $X_{n+1} = (aX_n) \bmod (2^{31} - 1)$
- تعداد کمی a در شرایط صدق می‌کنند $a = 7^5 = 16807$
- کامپیوترهای IBM360

مولد خطی هم‌نهشتی (ادامه)

- با انتخاب مناسب مقادیر a, c, m دنباله اعداد به دست آمده مشابه یک انتخاب تصادفی از مجموعه $\{1,2,...,m-1\}$ می‌باشد
- الگوریتم یقینی است! تنها قسمت تصادفی انتخاب مقدار اولیه X_0 است
- اطلاعات دشمن: نوع الگوریتم و پارامترها $a = 7^5, c = 0, m = 2^{31} - 1$
  - با مشخص شدن یک عدد، تمامی اعداد بعدی مشخص می‌شوند
- اطلاعات دشمن: نوع الگوریتم
  - دانستن طول محدودی از دنباله برای تعیین پارامترها کافی است

$$X_1 = (aX_0 + c) \bmod m$$
$$X_2 = (aX_1 + c) \bmod m$$
$$X_3 = (aX_2 + c) \bmod m$$

- استفاده از منابع تصادفی
  - دنباله پس از هر N عدد، با مقدار لحظه‌ای کلاک سیستم باز آغاز شود

مولد Blum Blum Shub (۱۹۸۶)

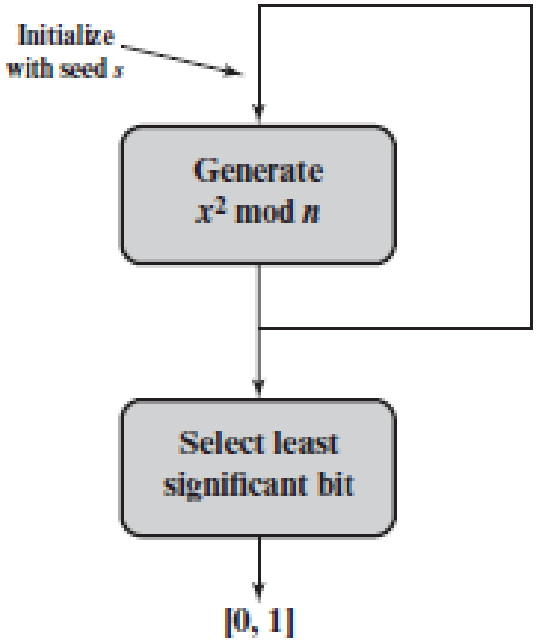
- دو عدد اول بزرگ p و q را انتخاب می‌کنیم، به طوری که:

$$p \equiv q \equiv 3 \bmod 4$$

$$n = p \times q$$

$$s : \text{random} \rightarrow (s, n) = 1$$

$$X_0 = s^2 \bmod n$$
$$\textbf{for } i = 1 \textbf{ to } \infty$$
$$X_i = (X_{i-1})^2 \bmod n$$
$$B_i = X_i \bmod 2$$



Blum Blum Shub مولد

$$n = 192649 = 383 \times 503$$

$$s = 101355$$

i	X_i	B_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0
10	177046	0

i	X_i	B_i
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	137171	1
20	48060	0

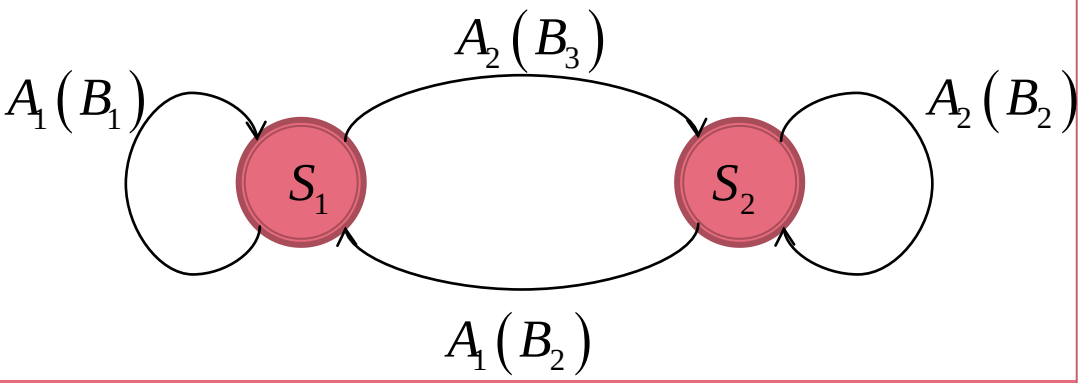
مولد Blum Blum Shub

- مولد بیت شبه تصادفی امن از نظر رمزنگاری
- Cryptographically secure pseudorandom bit generator (CSPRBG)
- تست بیت بعدی
  - با دانستن k بیت قبلی، احتمال صفر یا یک بودن بیت بعدی ۰.۵ باشد
    - در زمان چندجمله‌ای
  - غیرقابل پیش‌بینی از نظر محاسباتی
    - پیچیدگی معادل با تجزیه n به دو عامل اول

ماشین حالت محدود (finite state machine)

- مجموعه حالات محدود $S=\{S_i\}$
- الفبای محدود ورودی $A=\{A_i\}$
- الفبای محدود خروجی $B=\{B_i\}$
- خروجی تابعی از ورودی و حالت ماشین $B_k=\mu(A_i,S_j)$
- حالت بعدی تابعی از ورودی و حالت قبلی $S'_j=\delta(A_i,S_j)$
- دیاگرام حالت

حالت اولیه	ورودی	A_1	A_2
S_1		B_1,S_1	B_3,S_2
S_2		B_2,S_1	B_2,S_2

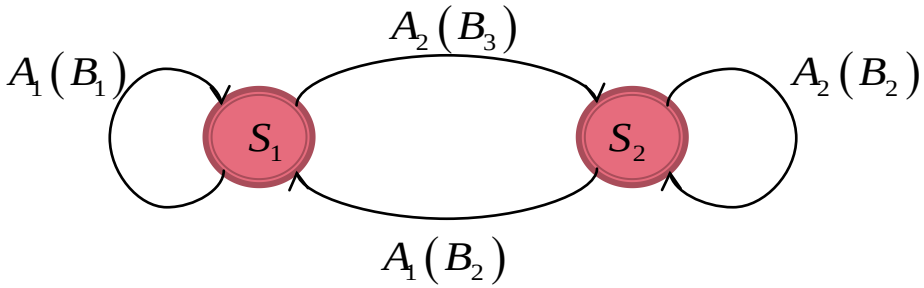


ماشین حالت محدود (ادامه)

$$\overbrace{A_1A_1A_2}\ \overbrace{A_1A_1A_2}\ A_1A_1A_2\ \ldots$$

$$S_1\overbrace{S_1S_1}\ S_2\overbrace{S_1S_1}\ S_2S_1S_1\ \ldots$$

$$B_1\overbrace{B_1B_3}\ B_2\overbrace{B_1B_3}\ B_2B_1B_3\ldots$$

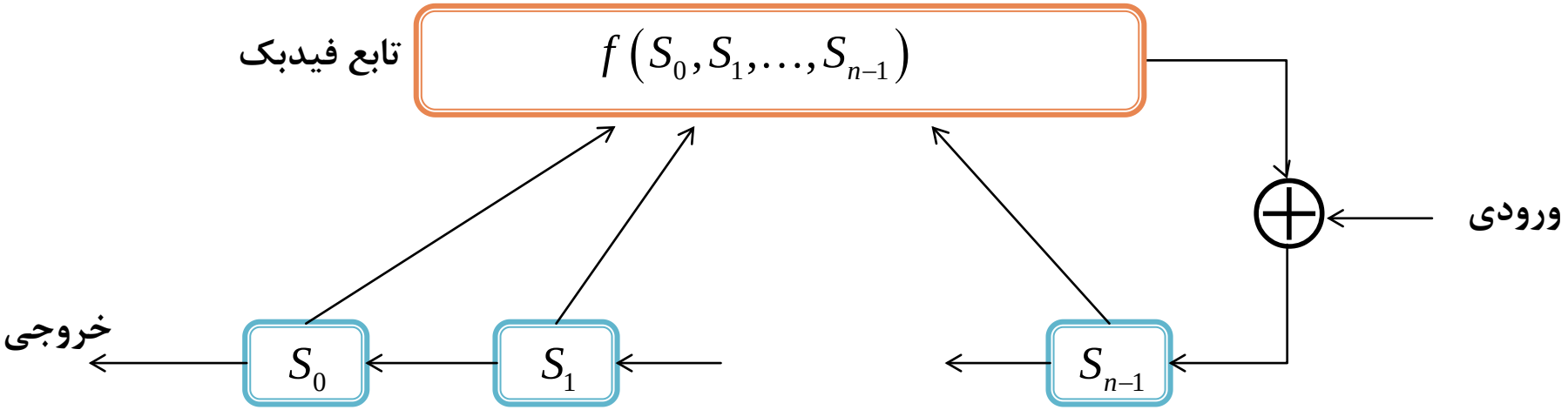


- اگر دنباله ورودی یک ماشین حالت محدود، یک دنباله در نهایت متناوب باشد، دنباله خروجی نیز در نهایت متناوب است

- اگر ماشین حالت محدود، الگوریتم تولید کلید باشد، دنباله کلید تولید شده یک دنباله در نهایت متناوب خواهد بود
  - اگر شیفت رجیستر با فیدبک خطی: دنباله از ابتدا متناوب
  - ارضای A1 و A2

شیفت رجیستر

- مجموعه‌ای از حافظه‌های باینری



- تعداد حالت‌ها $\|S\| = 2^n$
  - ماشین حالت محدود با الفبای ورودی و خروجی باینری

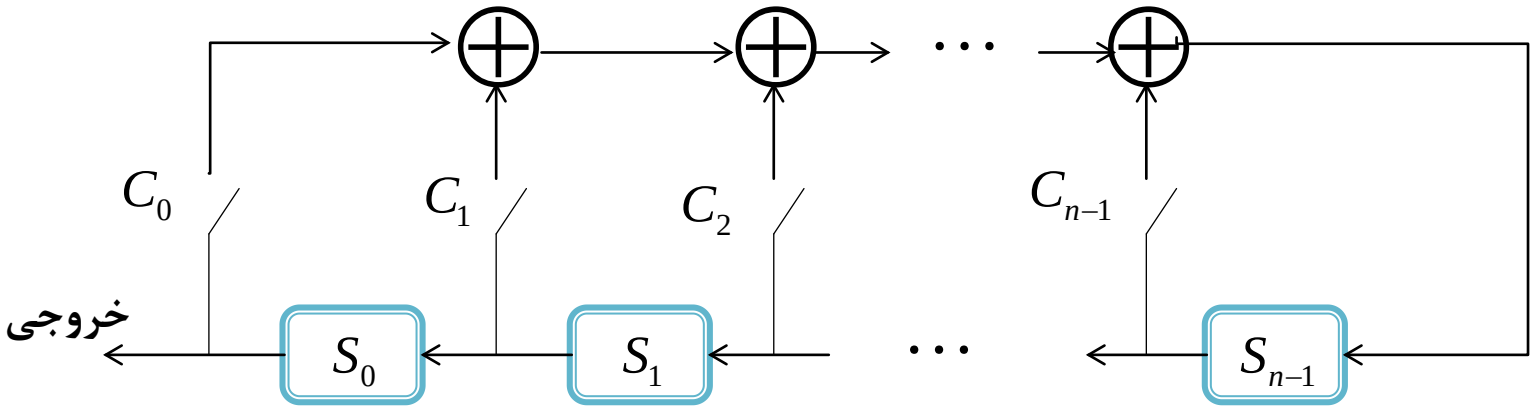
شیفت رجیستر با فیدبک خطی (LFSR)

- تابع فیدبک خطی باشد

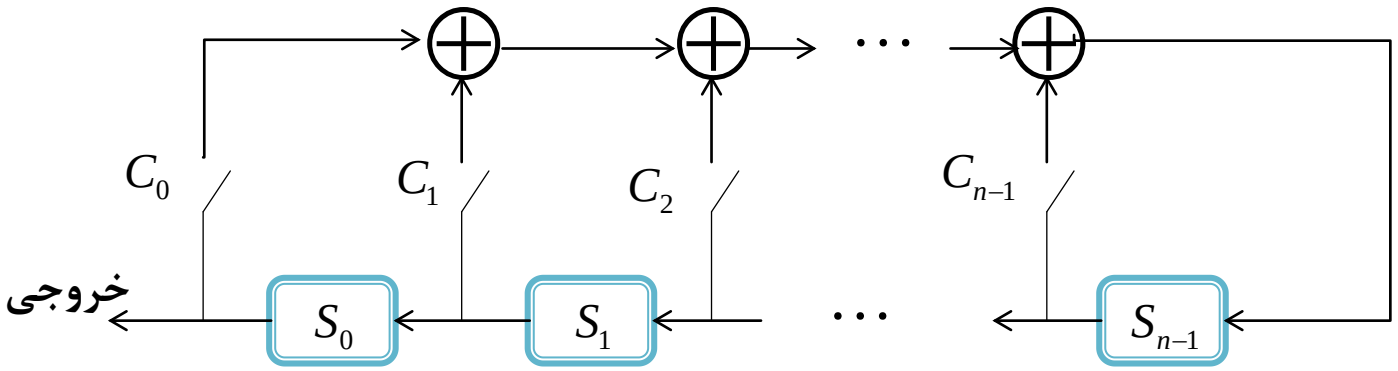
$$f(S_0, S_1, \ldots, S_{n-1}) = C_0 S_0 + C_1 S_1 + \ldots + C_{n-1} S_{n-1}$$

$$C_i \in \{0,1\}$$

$$C_0 = 1$$



شیفت رجیستر با فیدبک خطی (LFSR)



- خروجی یک LFSR با ورودی صفر و فرض $C_0 = 1$ یک دنباله متناوب است
  - n طبقه: دوره تناوب $p \le 2^n - 1$
  - اگر دوره تناوب حداکثر شود: M-sequence (Maximal length sequence)
  - چندجمله‌ای مشخصه $f(x) = 1 + C_1 x + \ldots + C_{n-1} x^{n-1} + x^n = \sum_{i=0}^{n} C_i x^i$

شیفت رجیستر با فیدبک خطی (LFSR)

- چندجمله‌ای‌ها در $\mathrm{GF}(2^n)$ $f(x)=1+C_1x+\ldots+C_{n-1}x^{n-1}+x^n=\sum_{i=0}^{n}C_ix^i$

$C_0=1$

- چندجمله‌ای ساده‌نشدنی (irreducible)
  - هرگاه بر هیچ چندجمله‌ای به جز ۱ و خودش بخش‌پذیر نباشد
- نما (exponent)
  - کوچکترین عدد صحیح مانند e به طوری‌که: $f(x)\mid x^e+1 \quad \Leftarrow \quad e\le 2^n-1$
- چندجمله‌ای اولی (primitive)
  - هر چندجمله‌ای ساده‌نشدنی با نمای بیشینه $e=2^n-1$
- شرط لازم و کافی برای آن که خروجی LFSR دوره تناوب بیشینه داشته باشد، آن است که **چندجمله‌ای مشخصه اولی** باشد

$p=2^n-1$

- A1

شیفت رجیستر با فیدبک خطی (LFSR)

- LFSRای که چندجمله‌ای مشخصه‌اش اولی باشد، معیارهای سه‌گانه گالوب را برآورده می‌کند
  - A2 ✓

$$f(x)=1+C_1x+\ldots+C_{n-1}x^{n-1}+x^n=\sum_{i=0}^{n}C_ix^i$$

- تعداد چندجمله‌ای‌های اولی از درجه n:

$$\begin{cases}\lambda(n)=\dfrac{\phi\left(2^n-1\right)}{n}\rightarrow\lambda(n)=\dfrac{2^n-2}{n}\\ 2^n-1:\text{prime}\end{cases}$$

  - تعداد چندجمله‌ای‌های درجه n: 2^n
  - برای nهای بزرگ تفاوت چندانی نمی‌کند

$$n=128\rightarrow 2^{128}\approx 10^{39}\ ,\ \frac{2^{128}}{128}\approx 10^{37}$$

امنیت رمز دنباله‌ای با LFSR

- کلید = حالت اولیه (n بیت) + ضرایب چندجمله‌ای فیدبک (n-1 بیت)
  - حالت اولیه به جز صفر
  - چندجمله‌ای اولی

$$\|K\| = \lambda(n)\left(2^n - 1\right) = \frac{\left(2^n - 1\right)\phi\left(2^n - 1\right)}{n} \simeq \frac{2^n \cdot 2^n}{n} = \frac{2^{2n}}{n}$$

$$H(K) = \log\left(\frac{2^{2n}}{n}\right) \simeq 2n \rightarrow N_0 \simeq \frac{2n}{3.2} = 0.6n$$

- با افزایش n
  - در برابر حمله نوع اول مقاوم است

امنیت رمز دنباله‌ای با LFSR

- حمله نوع دوم
  - متن اصلی متناظر با بخشی از متن رمزشده در اختیار است

$$p_i \oplus k_i = c_i \rightarrow k_i = p_i \oplus c_i$$

- تعداد معینی از بیت‌های دنباله کلید در اختیار است
  - تعداد معینی از بیت‌های دنباله خروجی LFSR در اختیار است
  - هدف: بدست آوردن کلید = حالت اولیه (n بیت) + ضرایب چندجمله‌ای فیدبک (n-1 بیت)
- چند بیت (متوالی) از دنباله کلید معلوم باشد تا سیستم شکسته شود؟ $2l \leq 2n$
  - چندجمله‌ای اولی: $2l = 2n$

امنیت رمز دنباله‌ای با LFSR

$n = 64 \rightarrow p = 2^{64} - 1 \simeq 10^{20}$

$2n = 128$

- A1: دوره تناوب دنباله کلید بزرگ
- A2: دنباله کلید تولید شده تصادفی به نظر برسد
  - معیارهای گالوب
- $2n$-1 مجهول
  - با $2n$ معادله **خطی** به جواب می‌رسد
  - A3: الگوریتم تولید کلید غیرخطی باشد
- ساختار غیرخطی
  - تابع فیدبک غیرخطی (شیفت رجیستر با فیدبک غیرخطی (NLFSR))
  - ترکیب کننده‌های غیرخطی (میان چند LFSR)

جدول صحت (truth table)

ورودی			خروجی	تابع سطر
S_0	S_1	S_2	S_3	
0	0	0	1	$(S_0+1)(S_1+1)(S_2+1)$
0	0	1	1	$(S_0+1)(S_1+1)S_2$
0	1	0	0	$(S_0+1)S_1(S_2+1)$
0	1	1	1	$(S_0+1)S_1S_2$
1	0	0	0	$S_0(S_1+1)(S_2+1)$
1	0	1	0	$S_0(S_1+1)S_2$
1	1	0	1	$S_0S_1(S_2+1)$
1	1	1	0	$S_0S_1S_2$

$$S_0S_1S_2 \rightarrow S_1S_2S_3$$

- تابع سطر: با جاگذاری مقادیر سطر، یک شود
- تابع فیدبک: مجموع حاصلضرب تابع هر سطر در خروجی همان سطر

$$\begin{aligned} f(S_0,S_1,S_2) &= (S_0+1)(S_1+1)(S_2+1)\cdot 1 \\ &+ (S_0+1)(S_1+1)S_2\cdot 1 \\ &+ (S_0+1)S_1(S_2+1)\cdot 0 + \cdots \\ &= 1 + S_0 + S_1 + S_1S_2 \end{aligned}$$

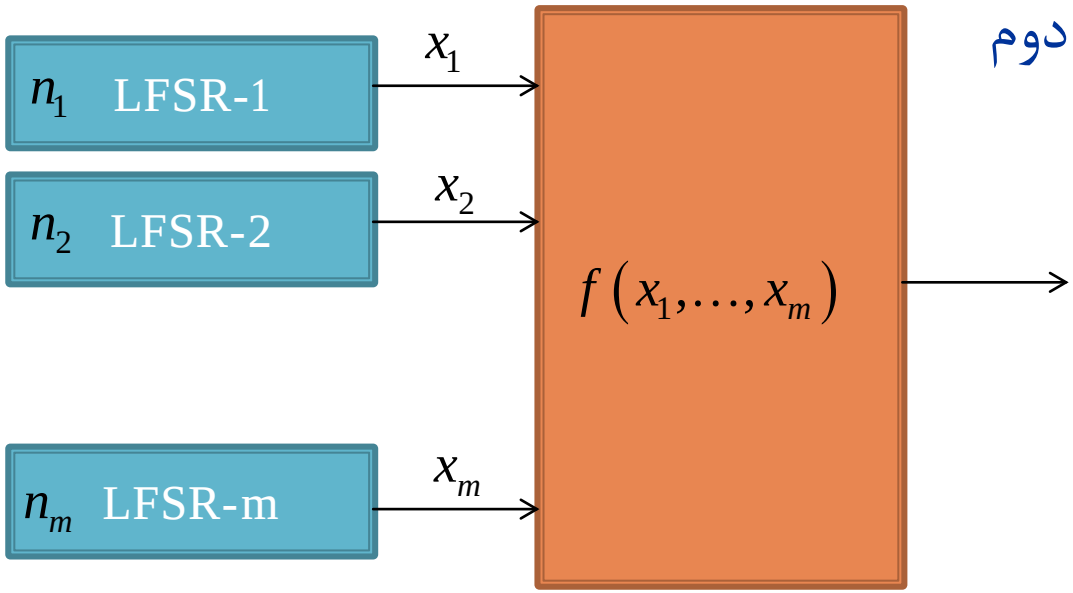
شیفت رجیستر با فیدبک غیرخطی (NLFSR)

- هر جدول صحت یک تابع فیدبک را می‌دهد و برعکس
  - تناظر یک به یک
  - تعداد توابع فیدبک = تعداد جداول صحت = 2^{2^n}
  - تعداد توابع فیدبک خطی = 2^n
- برای شکستن در حمله نوع دوم 2×2^n بیت لازم است
- تحت شرایطی!

ترکیب کننده‌های غیرخطی

$$\left(n_i, n_j\right)=1$$

- قضیه Ruppel-Staffelbach
  - چندجمله‌ای‌ها اولی

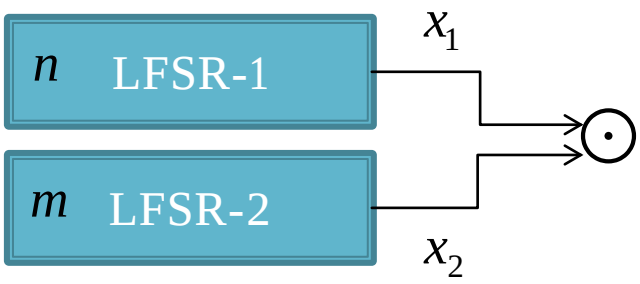


- بیت مورد نیاز در حمله نوع دوم

$$2\times f\left(n_1, n_2, \ldots, n_m\right)$$

ترکیب کننده‌های غیرخطی

- ترکیب کننده Hadamard

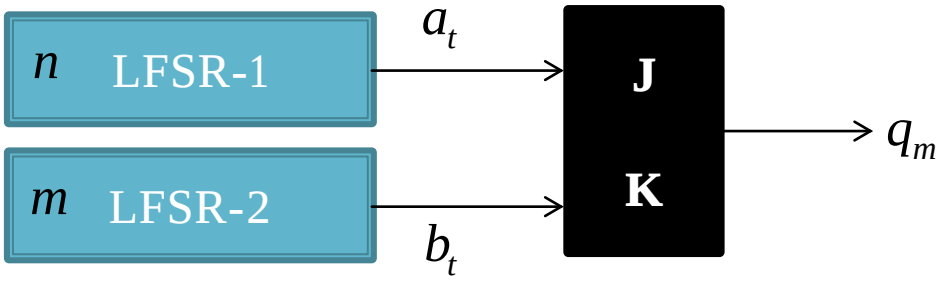


- بیت مورد نیاز در حمله نوع دوم $2 \times nm$

- احتمال بیت ۱: $\frac{1}{4}$
  - معیار R1 را بر آورده نمی‌کند
  - با دانستن یک بیت $x_3 = 1$ دو بیت از LFSR ها مشخص می‌شود
  - هم‌بستگی دنباله خروجی با تک تک ورودی‌ها

ترکیب کننده‌های غیرخطی

- ترکیب کننده حالت به کمک flip-flop



J	K	q_n
0	0	q_{n-1}
0	1	0
1	0	1
1	1	$\overline{q_{n-1}}$

$$q_n = (a_n + b_n + 1)q_{n-1} + a_n$$

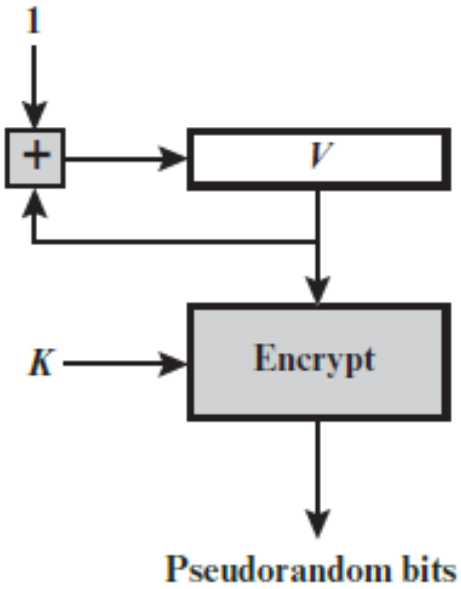
$$p = (2^m - 1)(2^n - 1)$$

- معیارها و خواص آماری

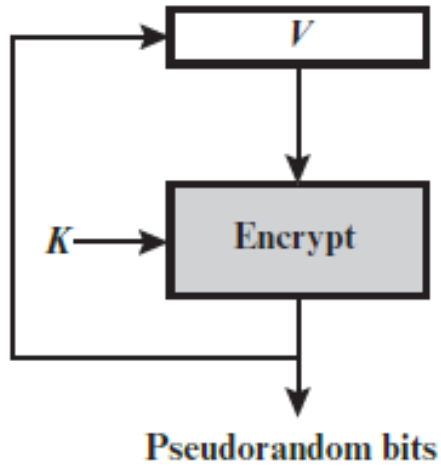
- هر ۲ بیت متوالی یک بیت LFSR را می‌دهد

تولید دنباله شبه تصادفی بر پایه رمزهای قالبی

- الگوریتم‌های رمزنگاری ورودی را تصادفی می‌کنند
- کاربرد: سیستم‌هایی که از الگوریتم خاصی برای رمزنگاری استفاده می‌کنند
  - سبک کاری CTR: استانداردهای NIST SP 800-90A، ANSI X9.82، RFC 4086
  - سبک کاری OFB: X9.82، RFC 4086



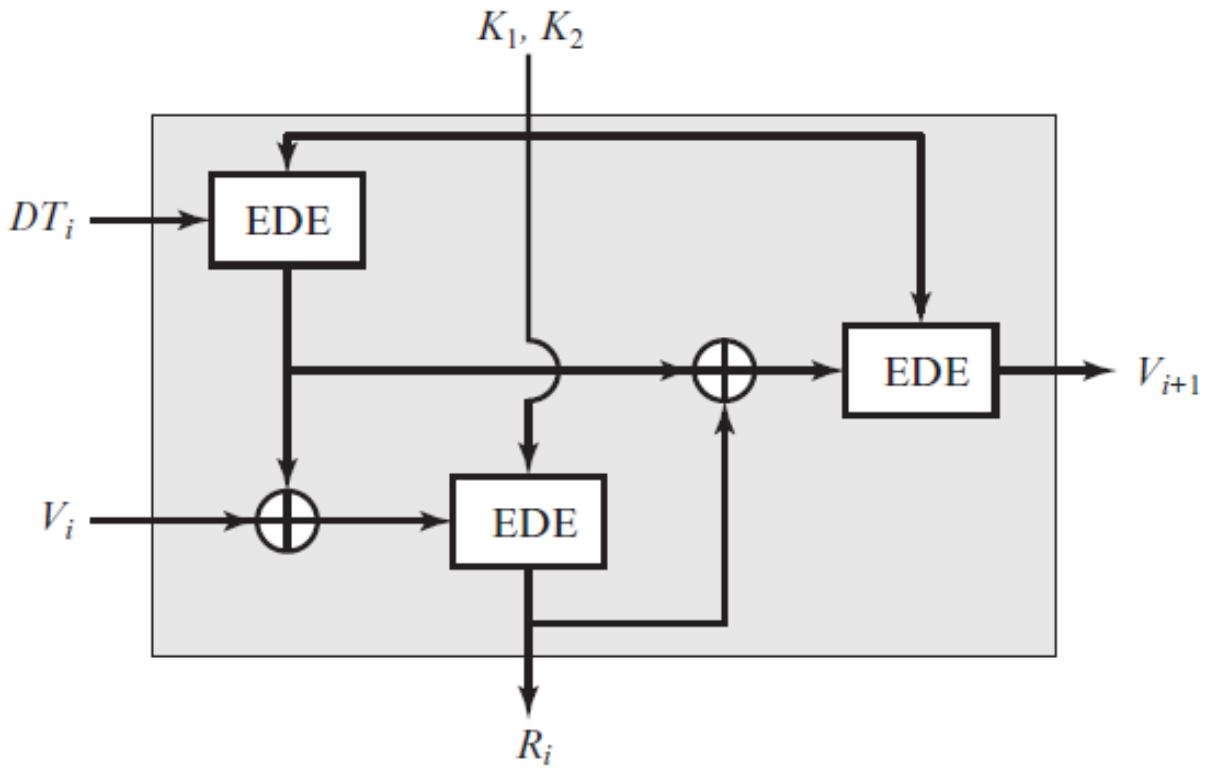
(a) CTR mode



(b) OFB mode

ANSI X9.17 PRNG

- کاربردهای امنیت سیستم‌های مالی و PGP
- ۳ ماجول 3-DES



NIST SP 800-90A - CTR_DRBG

- counter mode–deterministic random bit generator
- پروسسورهای اینتل
- استفاده از یک منبع تصادفی فیزیکی و TRNG
- استفاده از 3DES با ۳ کلید یا AES

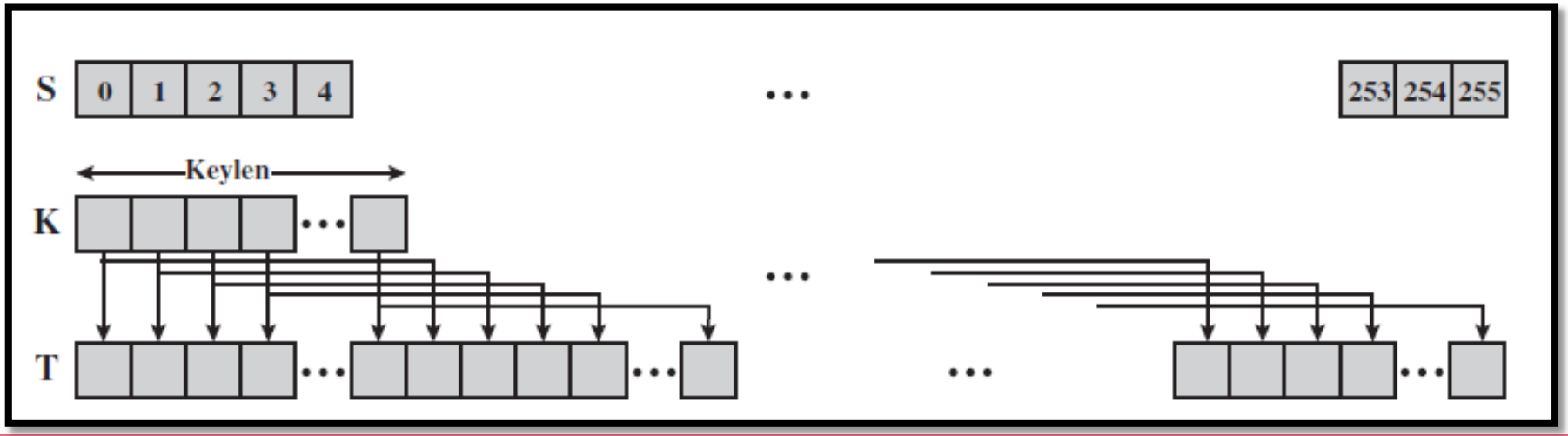
RC4

- توسط Ron Rivest در ۱۹۸۷ برای امنیت سیستم RSA
  - رمز دنباله‌ای با طول کلید متغیر و عملیات بایتی
  - بر پایه جایگشت تصادفی
  - دوره تناوب دنباله کلید بیشتر از ۱۰۱۰۰
  - سرعت زیاد
- استفاده در امنیت 802.11 wireless LAN
  - پروتکل Wired Equivalent Privacy (WEP) و پروتکل WiFi Protected Access (WPA)
- استفاده در پروتکل Kerberos و امنیت لایه انتقال (Secure Shell (SSH)
- استفاده در امنیت وب
  - استاندارد Secure Sockets Layer/Transport Layer Security (SSL/TLS)
- تا سال ۱۹۹۴ مخفی بود!
- کلید = ۱ تا ۲۵۶ بایت (۸ تا ۲۰۴۸ بیت)، بردار حالت=۲۵۶ بایت

RC4

1. مقداردهی اولیه (Initialization)
   - مقادیر ۰ تا ۲۵۵ در S ریخته می‌شوند $S[0] = 0, S[1] = 1, \ldots, S[255] = 255$
   - کلید در T کپی و به مقدار مورد نیاز تکرار می‌شود

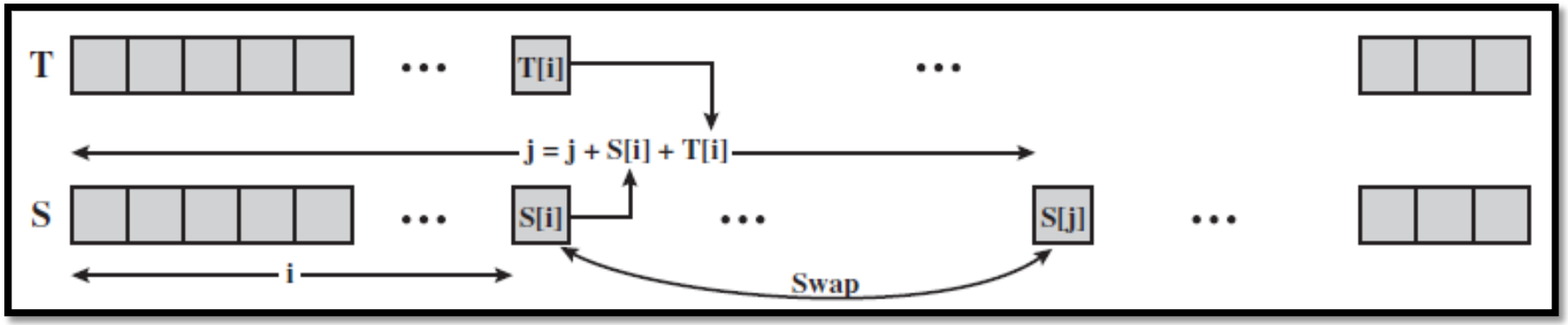
```
/* Initialization */
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```



RC4

2. جایگشت اولیه (Initial permutation of S)
   - هر بایت در S[i] را با بایت دیگری از آن جایگزین کن
   - این عمل را مطابق با مقدار T[i] انجام بده

```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
Swap (S[i], S[j]);
```

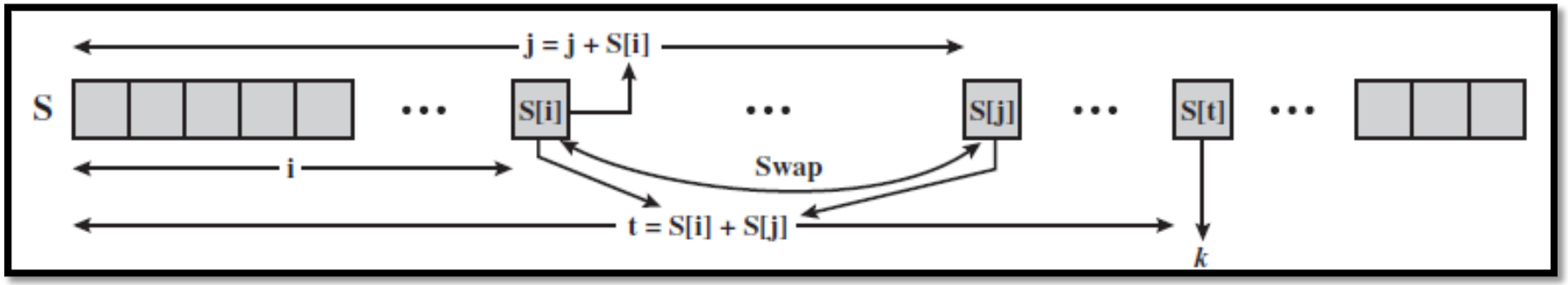


RC4

```
/* Stream Generation */
i, j = 0;
while (true)
  i = (i + 1) mod 256;
  j = (j + S[i]) mod 256;
Swap (S[i], S[j]);
t = (S[i] + S[j]) mod 256;
k = S[t];
```

3. تولید دنباله
   - کلید از الگوریتم خارج می‌شود
   - جایگشت S بر طبق مقادیر خودش

- رمزگذاری و رمزگشایی: XOR بایتی



امنیت RC4

- حملات زیادی به آن صورت گرفته
  - با طول کلید بیشتر از ۱۲۸ بیت ناموفق هستند

- مشکل در پروتکل WEP

- Fluhrer, S.; Mantin, I.; and Shamir, A. “Weakness in the Key Scheduling Algorithm of RC4.” *Proceedings,Workshop in Selected Areas of Cryptography,2001.*
  - استفاده در تامین محرمانگی 802.11 wireless LAN
  - آسیب‌پذیر
  - مشکل در RC4 نیست، بلکه در روش تولید کلید اصلی می‌باشد
  - در کاربردهای دیگر RC4 مشکلی نیست
  - سیستم امن = الگوریتم رمزنگاری + پروتکل

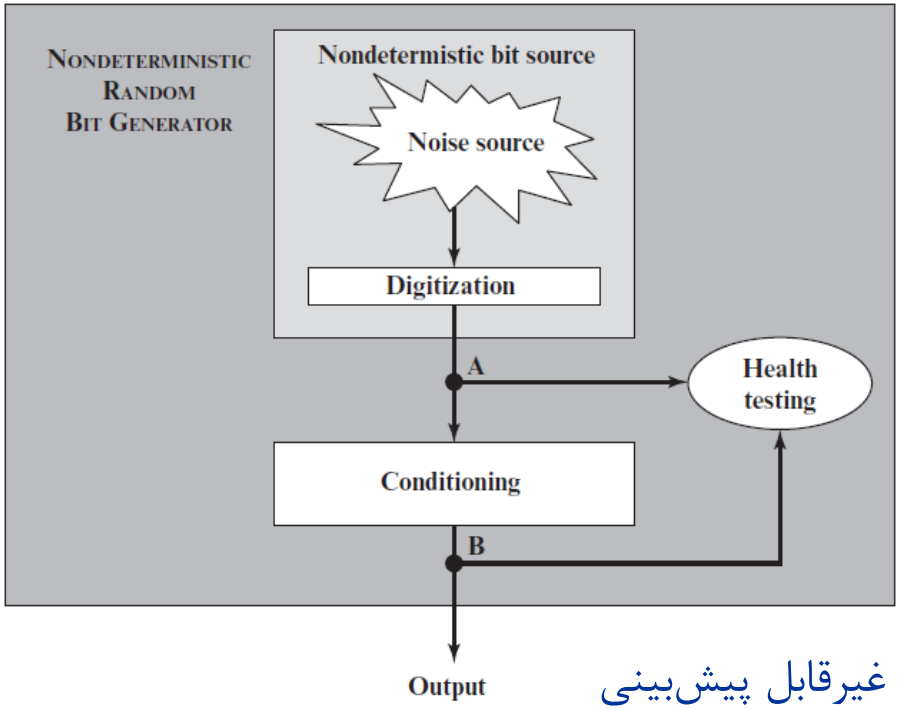
امنیت RC4

- تحلیل‌های اخیر نشان‌گر ضعف در دنباله کلید تولید شده:

- Paul, G., and Maitra, S. “Permutation after RC4 Key Scheduling Reveals the Secret Key”, *Selected Areas of Cryptography: SAC 2007, Lecture Notes on Computer Science*, Vol. 4876, pp. 360–337, 2007.
- Al Fardan, N., et al. “On the Security of RC4 in TLS and WPA.” *USENIX Security Symposium*, July 2013.

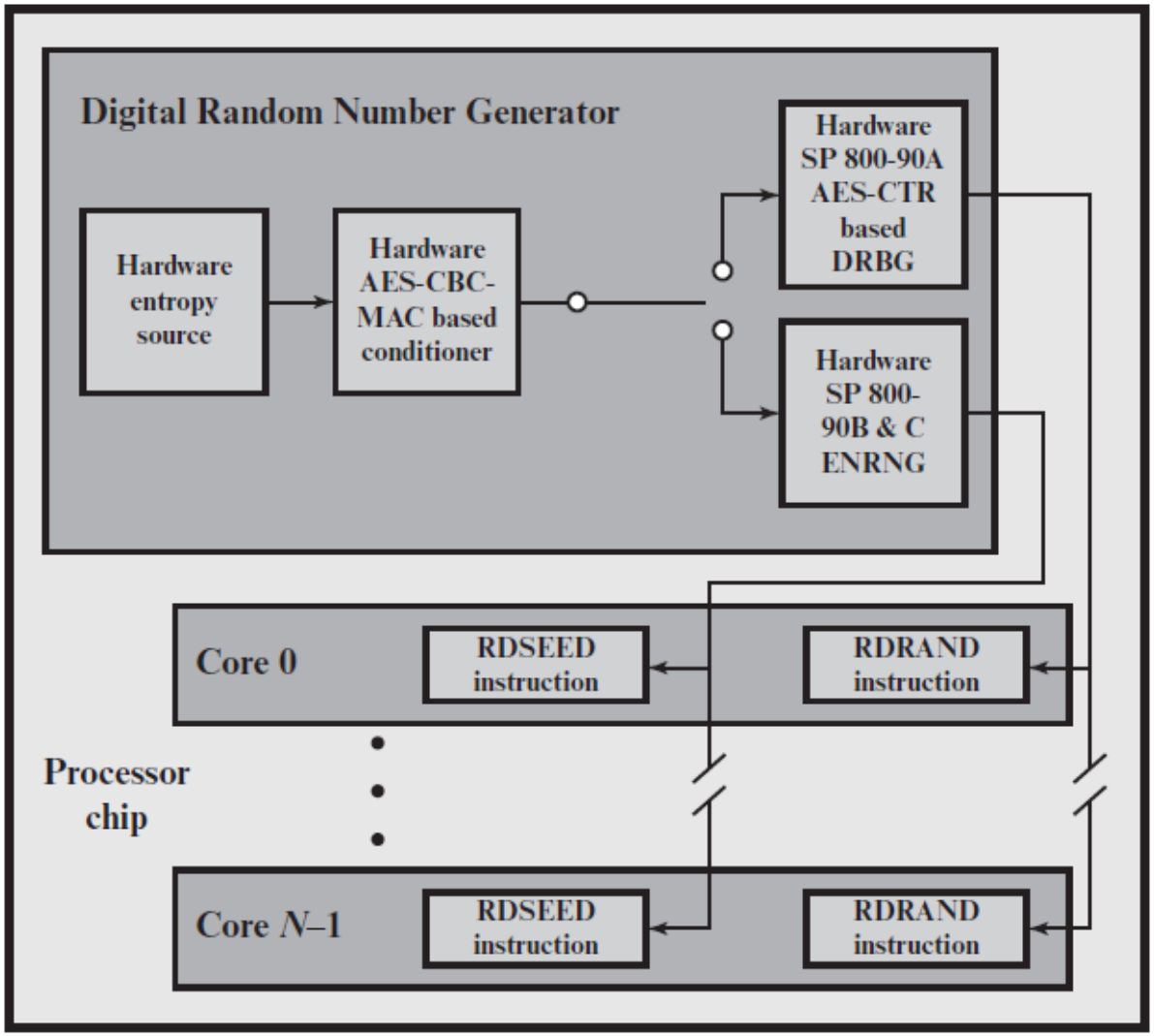
- منع استفاده در:
- TLS در RFC 7465 توسط IETF
  - *Prohibiting RC4 Cipher Suites*, February 2015
- کاربردهای دولتی توسط NIST
  - SP 800-52, Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations, September 2013

True random number generator (TRNG)



- استفاده از منبع تصادفی غیر یقینی
  - RFC 4086:
    - ورودی صدا/تصویر
    - دیسک درایو
- random.org
- NIST SP 800-90B
  - بایاس: توزیع نتایج یکنواخت نباشد
  - نرخ آنتروپی $0 \leq r \leq 1$: معیار تصادفی بودن و غیرقابل پیش‌بینی بودن
- Conditioning algorithms or deskewing algorithms
  - تابع چکیده‌ساز (Hash Function)
  - رمز قالبی (با کلید دلخواه)

Intel Digital Random Number Generator (DRNG)



- May 2012
- پیاده‌سازی سخت‌افزاری (امنیت و سرعت بالا)
- دنباله تصادفی با نرخ 4 Gbps (منبع آنتروپی سخت‌افزاری)
- مرحله آخر: شبه تصادفی
  - CTR_DRBG
  - ENRNG