مبانی رایانش امن
رمزنگاری- روش‌های متقارن- ذیل

محسن هوشمند
دانشکده تکنولوژی اطلاعات و علم رایانه
دانشگاه تحصیلات تکمیلی علوم پایه زنجان

ذیل رمزنگاری متقارن

نحوۀ تولید کلید یا نانس

نحوۀ ورودی بلوک داده

رمز دنباله

استفاده از اعداد تصادفی

لزوم استفاده از کلید و یا شمارنده در بخش‌های مختلف

پیش‌نیازهای متفاوت در مقادیر و ضرائب و عوامل مذکور

یکی از پیش‌نیازهای نانس‌ها و کلیدهای

- تولید تصادفی آنها
- نگاهی عامتر: رمزنگاری نیازمند تابع مهمی جهت تولید دنباله‌ای تصادفی از بیت‌ها
-

استفاده از اعداد تصادفی

لزوم استفاده از کلید و یا شمارنده در بخش‌های مختلف

پیش‌نیازهای متفاوت در مقادیر و ضرائب و عوامل مذکور

یکی از پیش‌نیازهای نانس‌ها و کلیدهای
- تولید تصادفی آنها
- نگاهی عامتر: رمزنگاری نیازمند تابع مهمی جهت تولید دنباله‌ای تصادفی از بیت‌ها

دو استراتژی متفاوت و کلی جهت تولید عدد تصادفی
- استراتژی نخست تولید قطعی و بیت‌ها با استفاده از الگوریتمی قطعی
  - به نام «مولد اعداد شبه‌تصادفی»
- استراتژی دیگر تولید نامعین بیت‌ها با استفاده از منبعی فیزیکی تولیدکنندهٔ خروجی تصادفی
  - دستهٔ متاخر «مولدهای راستین اعداد تصادفی» یا «مولد بیت تصادفی غیرقطعی»

استفاده از اعداد تصادفی

چند مورداستفاده

- توزیع کلید و احراز هویت متقابل
- تولید کلید جلسه
- تولید کلید برای رمزنگاری کلید عمومی رسا
- تولید دنباله بیت در رمزنگاری متقارن دنباله

استفاده از اعداد تصادفی

چند مورداستفاده

- توزیع کلید و احراز هویت متقابل
- تولید کلید جلسه
- تولید کلید برای رمزنگاری کلید عمومی رسا
- تولید دنباله بیت در رمزنگاری متقارن دنباله
- همگی دارای دو نیاز دو نیاز اساسی
  - «تصادفی‌بودن» و «پیش‌بینی‌ناپذیری»
  - لزوما با هم هم‌خوانی و برهم‌نهی ندارند
  - تصادفی‌بودن: تولید دنباله‌ای از اعداد تصادفی
  - دنباله اعداد دارای خصلت تصادفی در پارامترهای آماری خوش‌تعریف

استفاده از اعداد تصادفی

- وجود آزمون‌هایی برای تعیین انطباق دنباله‌ای از بیت‌ها با توزیعی خاص
- اما عدم وجود آزمونی جهت اثبات استقلال
  - به جای آن معرفی انواع آزمون‌ها جهت تشخیص عدم استقلال
  - استراتژی معمول
  - اعمال تعدادی از آزمون‌های عدم‌استقلال
  - تا دستیابی به اطمینان نسبتا بالا از استقلال داده‌ها
- در کاربردهایی مانند احراز متقابل و تولید کلید جلسه و رمز دنباله کافی نبودن تصادفی بودن آماری
  - نیاز به پیش‌بینی‌ناپذیری مقادیر بعدی دنباله
  - دنباله تصادفی واقعی: هر مقدار آمارا مستقل از مقادیر قبل و بعد

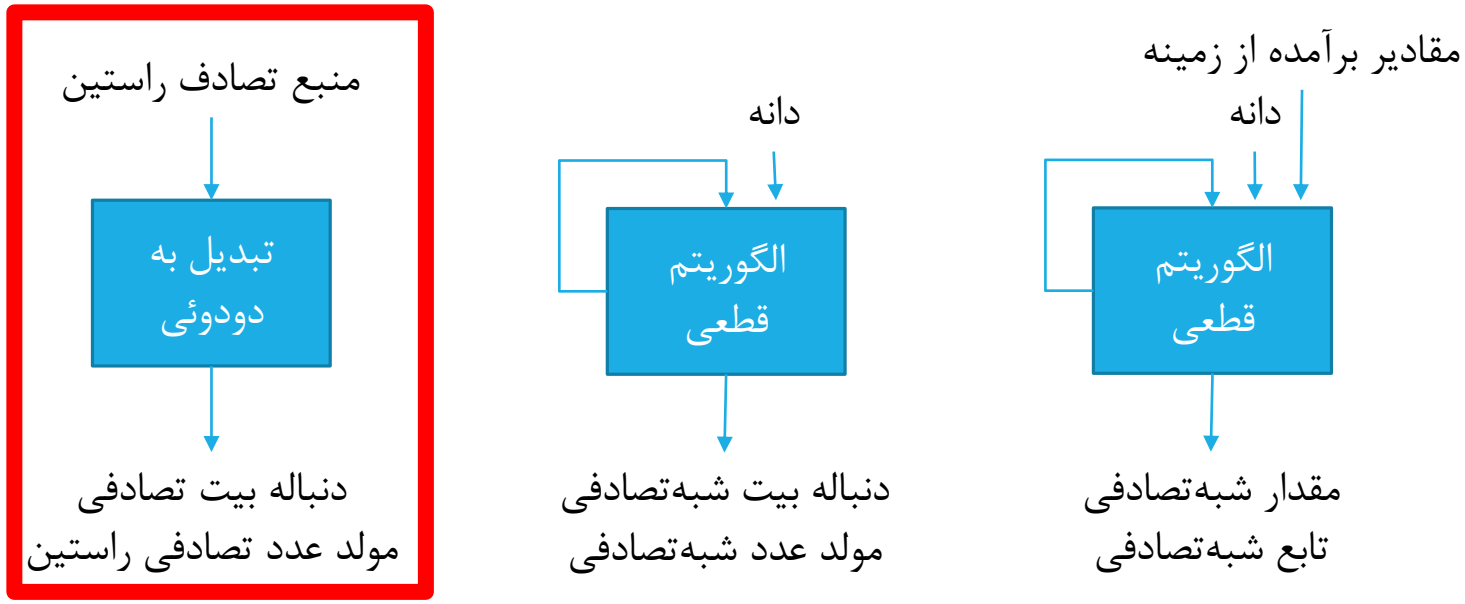
استفاده از اعداد تصادفی

- استفادهٔ کاربردهای رمزنگاری از صناعات الگوریتمی تولیدی اعداد تصادفی
- قطعی بودن الگوریتم‌های مذکور
  - آمارا تصادفی نبودن دنباله اعداد تولیدی
  - ولی با طراحی مناسب الگوریتم منجر به قبولی در آزمون‌های اعداد تصادفی
  - اعاداد حاصل معروف اعداد شبه‌تصادفی
  - مناسب بودن نسبی اعداد مذکور در عمل

اعداد تصادفی واقعی

مقایسه مولد اعداد تصادفی راستین با دو صورت تولید عدد شبه‌تصادفی

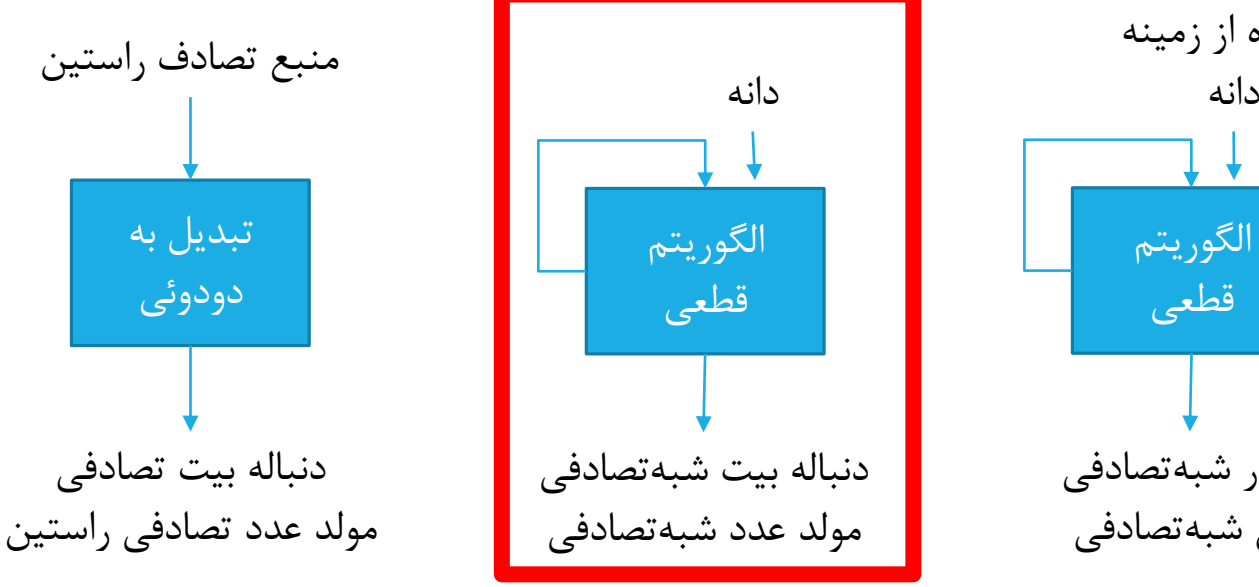
مولد اعداد تصادفی راستین-- منبع معمولا مشهور به منبع انتروپی



مولد اعداد شبه‌تصادفی

مولد اعداد شبه‌تصادفی–

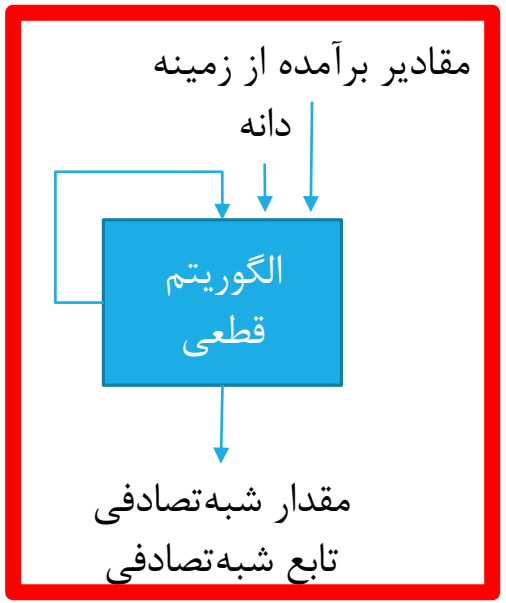
- ورودی ثابتی به نام دانه
- سپس تولید دنباله‌ای از بیت‌ها با الگوریتمی قطعیچ
- معمولا گرفتن دانه از اعداد تصادفی راستین
- مورد ویژه– مشخص شدن دنباله بیت خروجی با مقدار ورودی‌ها
  - یکسانی خروجی با یکسانی الگوریتم و دانه

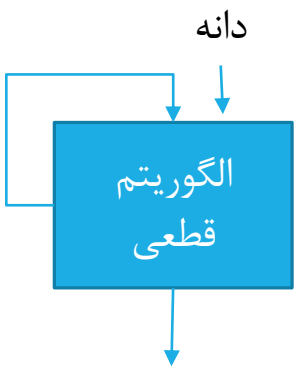


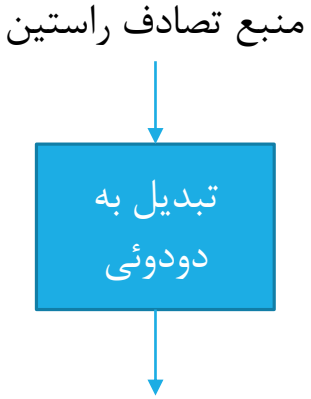
تابع شبه‌تصادفی

تابع شبه‌تصادفی–
- الگوریتمی برای تولید اعداد با پایان باز (طول ازاد)
- استفاده در مواردی چون رمز متقارن دنباله
- در قالب تولید تولید مقادیر شبه‌تصادفی با طول ثابت
  - استفاده در کلید رمزگذاری متقارن
  - نانس‌ها

تفاوت دو روش تصادفی: هیچی به جز طول کلید!







تصادفی بودن

پارامترهای لازم جهت بررسی تصادفی بودن

- یکنواختی
- مقیاس‌پذیری
- سازگاری

آزمون‌های بررسی تصادفی بودن

- آزمون بسامد
- آزمون اجراها
- آزمون جامع آماری مائر

پیش‌بینی‌ناپذیری

پیش‌بینی‌ناپذیری پیش‌رو

پیش‌بینی‌ناپذیری پس‌رو

دانه

نیاز به محرمانگی دانه

نیاز به پیش‌بینی‌ناپذیری دانه

در واقع خود باید عددی تصادفی یا شبه‌تصادفی

معمولا ایجاد دانه در منبع راستین تصادفی

- در صورت وجود منبع راستین چرا نیاز به الگوریتم‌های شبه‌تصادفی داریم؟

ایجاد الگوریتم

هدف‌محور

الگوریتم رمزگذاری‌محور

مولد عدد شبه‌تصادفی

دو نوع

- مولدهای هم‌ارزی خطی
- مولد بلوم بلوم شاب

مولد عدد شبه‌تصادفی-مولدهای هم‌ارزی خطی

بیشترین استفاده

لهمر ۱۹۵۱

دارای چهار ضریب

- پیمانه نامنفی m
- ضریب 0<a<m
- افزایش $0 \leq c < m$
- نقطهٔ عزیمت یا دانه $0 \leq X_0 < m$

تولید دنبالهٔ اعداد تصادفی X_n با تکرار معادلهٔ

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

در صورت مقدار صحیح بودن m، a، c، و X_0

امکان تولید دنباله عدد صحیح در بازهٔ 0 و m با الگوریتم

مولد عدد شبه‌تصادفی-مولدهای هم‌ارزی خطی

$$X_{n+۱} = (aX_n + c)\%m$$

اهمیت انتخاب مقادیر m، a، c در عملکرد الگوریتم

- مثال
  - مناسب نبودن دنباله در صورت $a = c = ۱$
  - تولید دنبالهٔ $\{۳,۷,۲۳,۱,۷\}$ با مقادیر $a = ۷$ و $c = 0$ و $m = ۳۲$ و $X_0 = ۱$
  - افزایش بازهٔ دنباله به طول هشت با تغیر a به ۵

مناسبتر بودن m-های بزرگ

- درتوانی تولید دنباله طولانی از اعداد تصادفی
- معمولا برابر با بزرگترین عدد صحیح نامنفی قابل نمایش

مولد عدد شبه‌تصادفی-مولدهای هم‌ارزی خطی

آزمون پارک جهت ارزیابی مولد عدد تصادفی
- تابع باید تابع مولد تمام بازه باشد.
- دنبالهٔ تولیدی باید تصادفی ظاهر شود.
- تابع باید کارآمدانه با حساب ۳۲ بیتی کار کند.

مورد نخست

با اول بودن m و $c = 0$ امکان اثبات اینکه برای بعضی از مقادیر a تولید بازه تابع تولید برابر با طول $m - ۱$

مثال

$$X_{n+۱} = (aX_n)\%(۲^{۳۱} - ۱)$$

- انتخابی معمول

در بیش از دو میلیارد انتخاب a، قبولی در هر سه آزمون صرفا ضرائب خاصی از a
- مثال $a = ۷^۵$
  - استفاده در خانوادهٔ آی‌بی‌ام ۳۶۰

مولد عدد شبه‌تصادفی-مولدهای هم‌ارزی خطی

در صورت

- دانستن استفاده از الگوریتم هم‌ارزی خطی
- دانستن ضرائب
- با مشخص شدن یکی از خروجی‌ها
  - ⟸ مشخص شدن کل خروجی‌های بعدی

یافتن ضرائب با صرفا اطلاع از استفاده از الگوریتم هم‌ارزی خطی و دیدن سه مقدار متوالی

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

حل با دستگاه سه معادله سه مجهوله

⟸ بنابرانی به دنبال روش سلب‌کنندهٔ حدس مقدار بعدی بر اساس مقدار فعلی

مولد عدد شبه‌تصادفی-مولد بلوم بلوم شاب

به نام معرفان این روش

محمتلا قوی‌ترین روش عمومی

در ابتدا انتخاب دو عدد اول بزرگ p و q

- به طوری‌که هر دو به پیمانه ۴ دارای باقیمانده ۳

$$p \overset{۴}{\equiv} q \overset{۴}{\equiv} ۳$$

- مثال صدق اعداد اول ۷ و ۱۱ در معادلهٔ بالا

تعریف $n = p \times q$

سپس انتخاب عددی تصادفی s با خاصیت اول بودن نسبت به n

- به دیگر سخن، مقسوم‌علیه نبودن p و q برای s نباشند.

مولد عدد شبه‌تصادفی-مولد بلوم بلوم شاب

$$p \equiv_{۴} ۳$$
$$q \equiv_{۴} ۳$$
$$n = p \times q$$
$$(s, n) = ۱$$

- الگوریتم ب‌ب‌ش
  - $X_0 = s^{۲} \% n$
  - $for\ i = ۱ : \infty$
    - $\mathrm{X_i} = \left(X_{i-۱}\right)^{۲} \% n$
    - $B_i = \mathrm{X_i} \% ۲$
- در واقع انتخاب کم‌ارزش‌ترین بیت در هر مرحله
- عدم امکان یافتن مقدار بعدی با احتمال بیشتر از نیم با هیچ الگوریتم زمان چندجمله‌ای در صورت انتخاب مناسب n

تولید عدد شبه‌تصادفی با استفاده از رمز بلوک

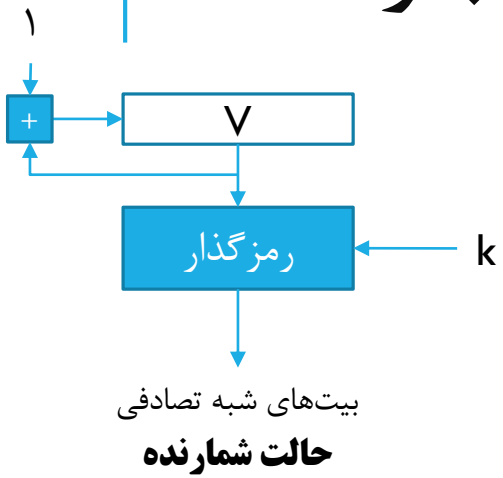
متن رمزی تولیدی ظاهرا تصادفی حاصل رمز بلوک متقارن به ازای هر متن ورودی

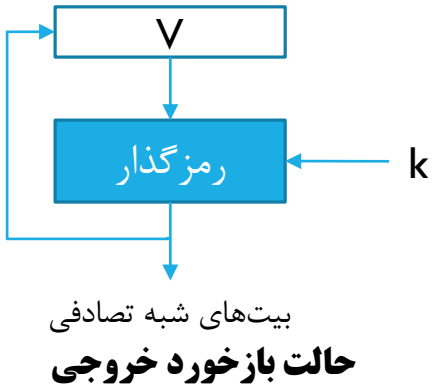
- نبود هیچ الگو یا مقادیری مشخص در متن
- امکان استفاده از خاصیت مذکور در تولید عدد تصادفی

دو حالت

- انتخاب بلوک شمارنده
- بازخورد خروجی

تولید عدد شبه‌تصادفی با استفاده از رمز بلوک

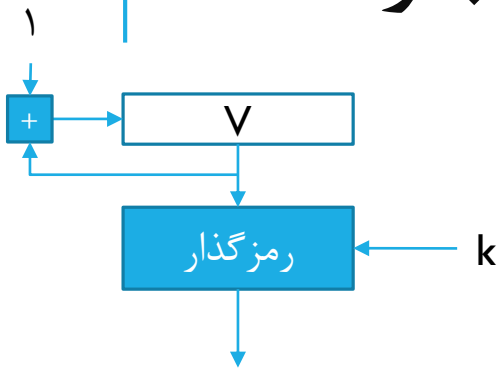




دانه دارای دو بخش

- مقدار کلید رمزنگاری
- مقدار V که پس از تولید هر بلوک عدد بروز می‌شود.
- بنابراین، $AES-۱۲۸$ دانه دارای کلید ۱۲۸ بیتی و مقدار V نیز ۱۲۸ بایتی
  - حالت شمارنده افزایش ۱ واحدی مقدار V در هر بار رمز
- در حالت بازخورد
  - خروجی V جدید برابر با مقدار خروجی عدد شبه‌تصادفی

تولید عدد شبه‌تصادفی با استفاده از رمز بلوک



حالت شمارنده

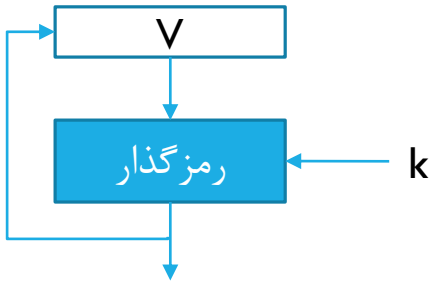
الگوریتم حالت شمارنده

while (len (temp) < requested_number_of_bits) **do**

$V = (V + ۱) \%۲^{۱۲۸}$

output_block = E(Key, V)

temp = temp || output_block



حالت بازخورد خروجی

الگوریتم حالت بازخورد خروجی

while (len (temp) < requested_number_of_bits) **do**

V = E(Key, V)

temp = temp || V

تولید عدد شبه‌تصادفی - انسی X9.17

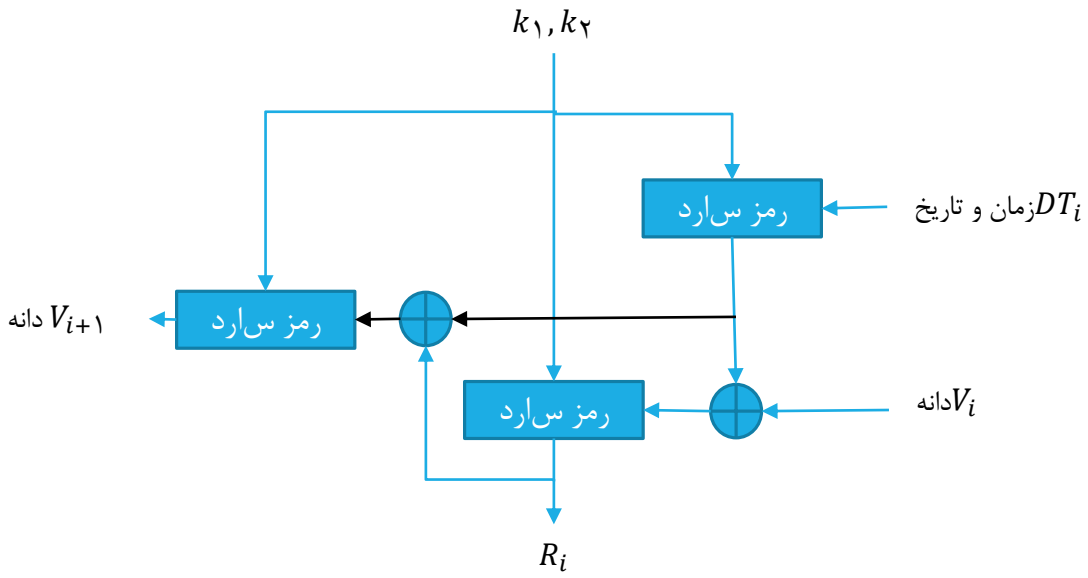
انسی X9.17

از قوی‌ترین روش‌های تولید عدد شبه‌تصادفی امنیت مالی

- مثال استفادهٔ PGP از روش مزبور

استفاده از رمزگذار س‌ارد

- ورودی‌های روش شامل
  - نمایش ۶۴ بیتی تاریخ و زمان فعلی
  - دیگری مقدار دلبخواه ۶۴ بیتی
- استفادهٔ هر سه س‌ارد از جفت یکسان کلیدهای ۵۶ بیتی
  - نیاز به محرمانه ماندن کلیدها
  - صرفا استفاده در تولید عدد تصادفی
- خروجی نیز شامل
  - عدد شبه‌تصادفی ۶۴ بیتی
  - دانه ۶۴ بیتی



انواع بخش‌بندی (بلوک‌بندی) داده‌ها جهت رمز

رمز بلوک دریافت
- بلوکی با طول ثابت فرضا b بیت
- کلید
- خروجی بلوک تحویل متن رمزی برابر با b بیت

درصورت بزرگتر از b بیت بودن طول دادهٔ دریافتی
- شکستن داده به بلوک‌های b بیتی

یادآوری
- امکان ایجاد مشکل امنیتی با رمزگذاری چند بلوک با کلیدی یکسان

راه‌حل
- تعریف ورود هر بلوک در پنج دسته عملیات (مد عملیات یا نحوهٔ عملیات)
- سازمان استاندارد ملی امریکا
- اعمال‌پذیری رمز بلوک روی چند بلوک داده یا دنباله‌ای از داده‌ها
- کاربرد در رمزنگاری‌های متقارن اعم از ارد و ارپ و جز این‌ها
- پنج مجموعه عملیات شامل
  - کتاب‌رمز الکترونیک، زنجیرهٔ بلوک رمز، بازخورد رمز، بازخورد خروجی، و شمارنده

انواع بخش‌بندی (بلوک‌بندی) داده‌ها جهت رمز

کتاب‌رمز الکترونیک

زنجیرهٔ بلوک رمز

بازخورد رمز

بازخورد خروجی

شمارنده

کتاب رمز الکترونیک

روش دفترچه رمز الکترونیک ساده‌ترین نحوه ممکن

- هر بار بخشی از داده‌ها در قالب بلوکی تحویل الگوریتم
- و دریافت متن رمز متناظر

دفترچه رمز

- زیرا هر متن اصلی b بیتی دارای متن رمز منحصربفردی با کلید داده شده
- جلوه چون دفترچه‌ای به ازای هر b بیت، خروجی خاصی
- تقسیم پیام‌های طولانی‌تر از b بیت به پیام‌هائی b بیتی
- لاگذاری پیام آخر در صورت لزوم

رمزگشائی

- در هر زمان بر یک بلوک رمز و همیشه با یک کلید
- در صورت تکرارb بیت از متن اصلی بیش از یک بار
  - متن رمز نیز یکسان

کتاب رمز الکترونیک

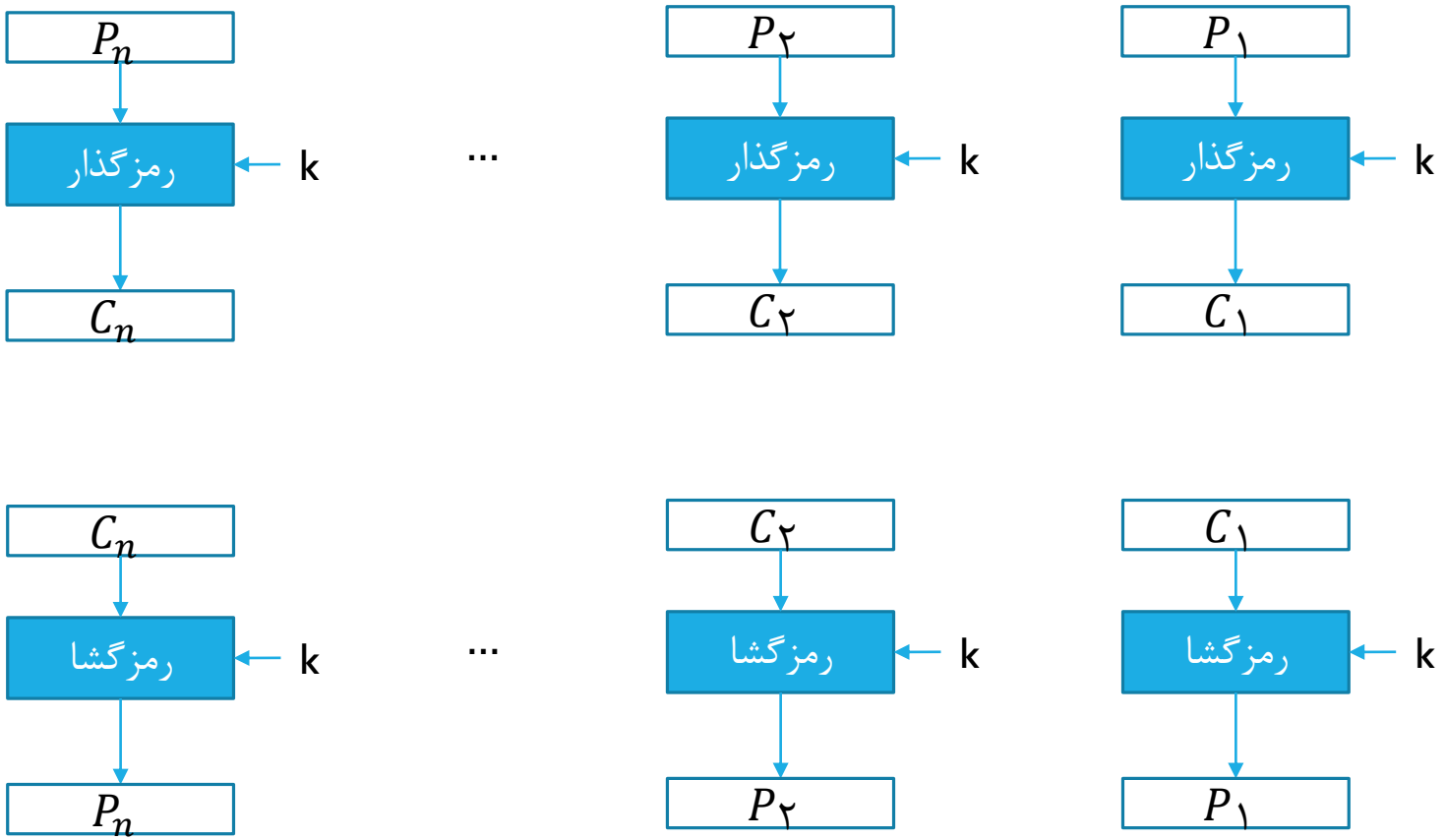
نامناسبی دفترچه برای رشته‌های طولانی

- معرفی روش‌هایی جهت افزودن پیچیدگی به بلوک داده
- استفاده از دفترچه رمز به عنوان پایهٔ مقایسه

معرفی چند معیار برای ارزیابی روش‌های مزبور

- شامل
- سربار- عملیات‌ها اضافی جهت رمزگذاری و رمزگشائی در مقایسه با رمزگذاری و رمزگشائی دفترچه رمز
- بازیابی خطا- خطائی در متن رمز i-ام صرفا در چند متن اصلی بعدی تکرار شود.
- انتشار خطا- خطائی در متن رمز i-ام صرفا در تمامی متن اصلی‌های بعدی باقی بماند.
  - منظور از خطا خطای ناشی از انتقال است و نه خطای محاسباتی.
- انتشار- چگونگی انعکاس آمارهای متن اصلی در متن رمز
  - جهت تقریبی سرانگشتی، انتروپی پائین برابر با پیش‌بینی‌پذیری یا عدم تصادفی بودن
- امنیت- بررسی درز اطلاعات بلوک متن اصلی با استفاده از بلوک متن رمز

کتاب رمز الکترونیک



زنجیره بلوک رمز

جهت غلبه بر معایب دفترچه رمز

- بدنبال روشی ایجاد متن رمز متفاوت برای متن بلوکی تکراری

زنجیره متن رمز

ورودی رمزگذار برابر با یاانحصاری بلوک متن اصلی فعلی و بلوک متن رمز قبلی

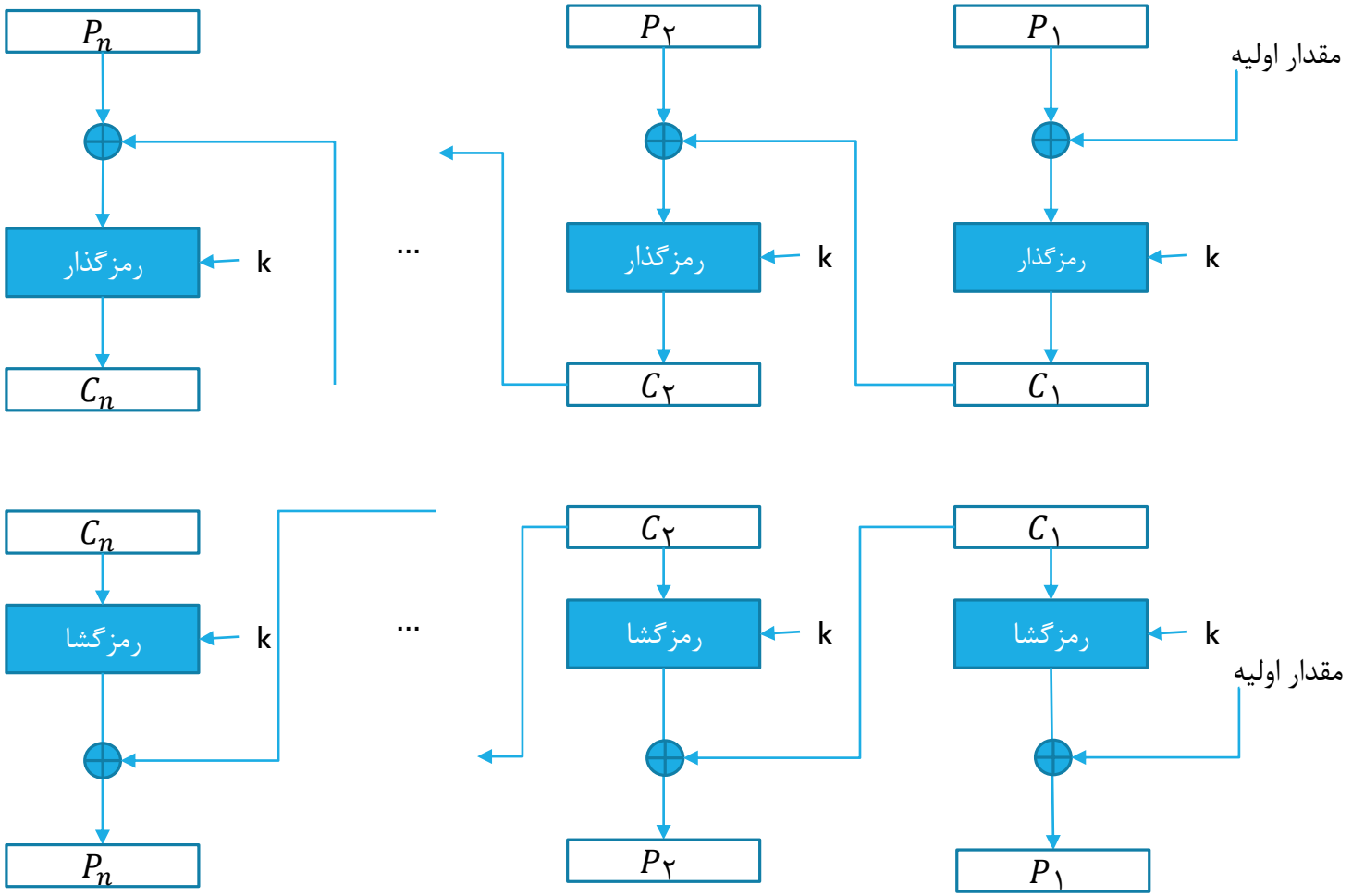
یکسانی کلید بلوک‌ها

- نبود ورودی هر تابع رمزگذاری دقیقا خود متن اصلی
- عدم‌بازنمائی تکرار الگوها
- نیاز به لاگذاری بلوک آخر همانند دفترچه رمز

زنجیره بلوک رمز

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

زنجیره بلوک رمز



زنجیره بلوک رمز

رمزگشایی نیز هر متن رمز از رمزگشا می‌گذرد

- یاانحصاری نتیجه با متن رمز قبلی

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

$$D(K, C_j) = D(K, E\left(K, \left[C_{j-1} \oplus P_j\right]\right) = C_{j-1} \oplus P_j$$

$$C_{j-1} \oplus D(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

زنجیره بلوک رمز

یاانحصاری مقدار اولیه‌ای با اولین متن اصلی در ابتدای کار

نیاز به مشخص بودن مقدار اولیه برای فرستنده و گیرنده

- همچنین عدم امکان پیش‌بینی شخص ثالث
- امکان ارسال مقدار اولیه با استفاده از دفترچه رمز

از دلایل حفاظت از مقدار اولیه

- درصورتی که شخص ثالث بتواند گیرنده را بفریبد
- استفاده از متن متفاوتی برای مقدار اولیه
  - قادر به برگرداندن متن اصلی انگاه قادر خواهد بود
  - امکان استفاده از نانس

استفاده در احراز هویت

حالت بازخورد رمز

اعمال روش‌های بلوکی روی بلوک‌ها

اما امکان تبدیل آنها به رمز دنباله با استفاده از سه روش بازخورد رمز و بازخورد خروجی و شمارنده

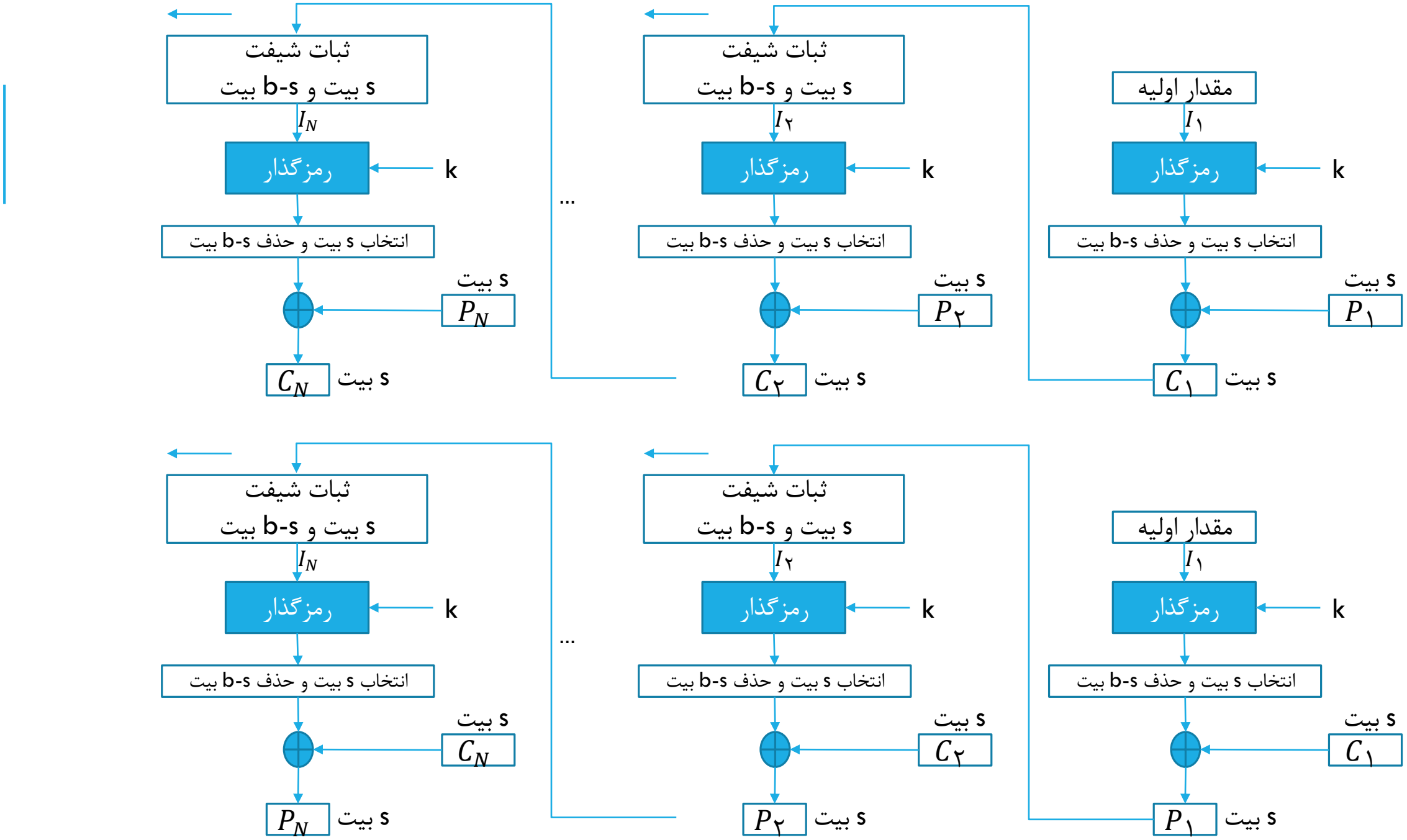
برطرف کردن نیاز به لاگذاری در رمز دنباله

- یکی شدن طول کل متن اصلی و رمز در رمزگذاری دنباله
- همچنین امکان اعمال بلادرنگ

فرض طول بیت‌های انتقالی برابر S بیت

- تقسیم متن اصلی به بخش‌هائی برابر S بیت

امکان انجام موازی محاسبات


مقدار اولیه
I_1
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_1
s بیت
C_1
ثبات شیفت
s بیت و b-s بیت
I_2
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_2
s بیت
C_2
...
ثبات شیفت
s بیت و b-s بیت
I_N
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_N
s بیت
C_N
مقدار اولیه
I_1
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_1
s بیت
P_1
ثبات شیفت
s بیت و b-s بیت
I_2
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_2
s بیت
P_2
...
ثبات شیفت
s بیت و b-s بیت
I_N
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_N
s بیت
P_N

حالت بازخورد خروجی

شبیه بازخورد رمز

خروجی تابع رمز به عنوان ورودی رمزگذاری بلوک بعدی متن اصلی بازخورد داده می‌شود

همچنین برروی کل بلوک متن اصلی و رمز کار می‌کند.

در حالی که بازخورد رمز روی s بیت اعمال می‌شود

$$C_j = P_j \oplus E(K, O_{j-۱})$$
$$O_{j-۱} = E(K, O_{j-۲})$$

با جایگذاری داریم:

$$C_j = P_j \oplus E(K, [C_{j-۱} \oplus P_{j-۱}])$$

با مرتب کردن و تنظیم متفاوت داریم:

$$P_j = C_j \oplus E(K, [C_{j-۱} \oplus P_{j-۱}])$$

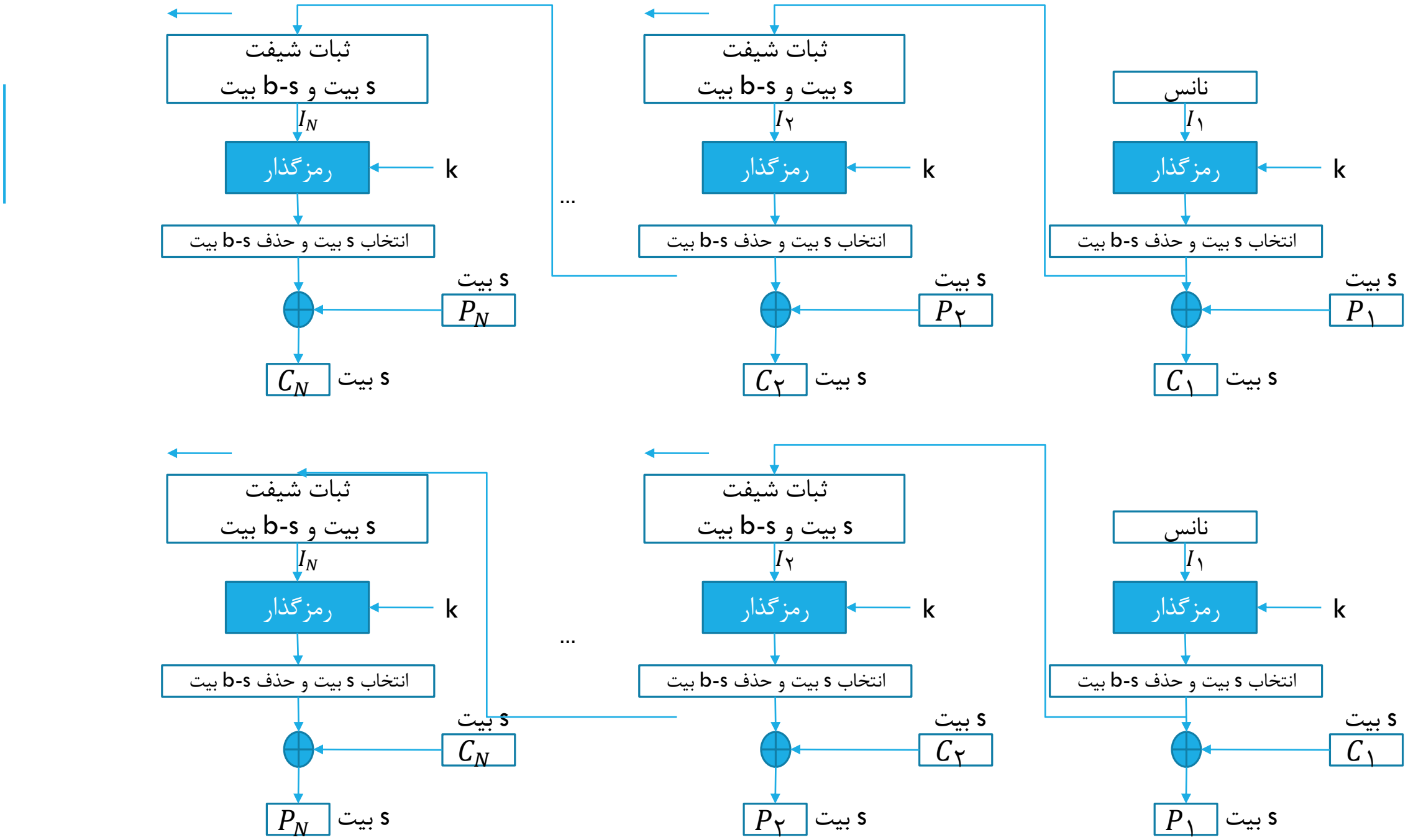
حالت بازخورد خروجی

نیاز به نانس

- به دلیل استفاده از مقدار آغازین در مرحله خروجی مراحل بعد
- عدم انتقال خطای بیتی در انتشار

ایراد

- تهدیدپذیری بیشتر روش مذکور به حمله تغییر پیام نسبت به روش بازخورد رمز


نانس
I_1
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_1
C_1
s بیت
ثبات شیفت
s بیت و b-s بیت
I_2
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_2
C_2
s بیت
...
ثبات شیفت
s بیت و b-s بیت
I_N
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
P_N
C_N
s بیت
نانس
I_1
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_1
P_1
s بیت
ثبات شیفت
s بیت و b-s بیت
I_2
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_2
P_2
s بیت
...
ثبات شیفت
s بیت و b-s بیت
I_N
رمزگذار
k
انتخاب s بیت و حذف b-s بیت
s بیت
C_N
P_N
s بیت

حالت شمارنده

سال ۱۳۵۸

استفاده از شمارنده‌ای برابر با بلوک متن اصلی

صرفا نیاز به تفاوت مقدار شمارنده از مقدار متن اصلی

معمولا مقدار اولیهٔ شمارنده مشخص می‌شود

- سپس افزایش یک واحدی به پیمانه ۲^b و b طول بلوک

یاانحصاری متن اصلی با شمارنده

عدم زنجیره‌سازی

حالت شمارنده

با داشتن دنباله‌ای از شمارنده‌های T_1 و T_2 و ... و T_n رمزگذاری شمارنده

$$C_j = P_j \oplus E(k, T_j)$$
$$C_N^* = P_N^* \oplus MSB_u(E(K, O_N))$$

رمزگشائی شمارنده نیز به صورت زیر است:

$$P_j = C_j \oplus E(k, T_j)$$
$$P_N^* = C_N^* \oplus MSB_u(E(K, O_N))$$

حالت شمارنده

شمارنده ابتدایی باید نانس

لزوم منحصر بفردی تمامی مقادیر شمارنده‌ها

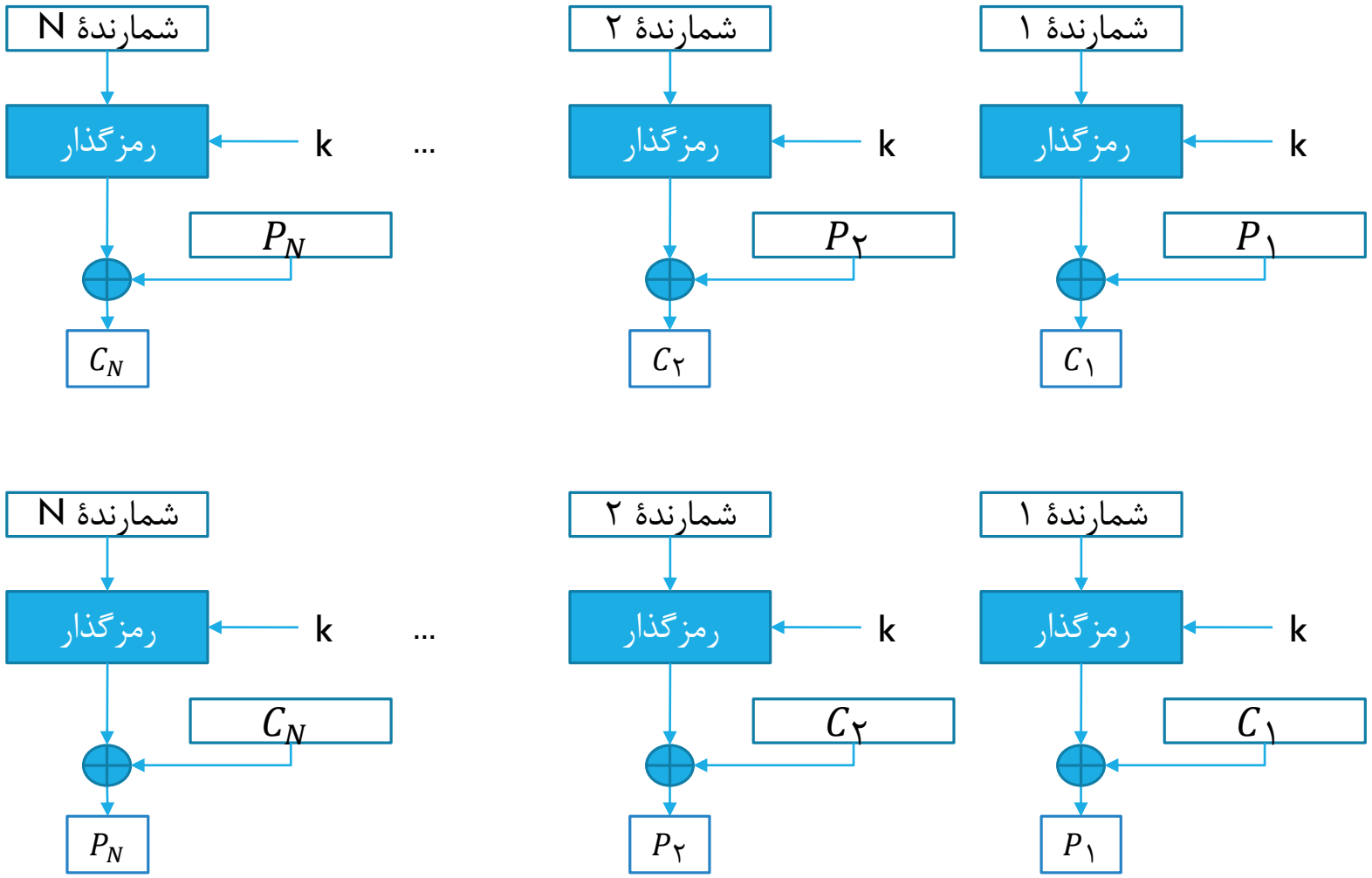
از روش‌های منحصر بفردی

- افزایش تک واحدی شمارنده

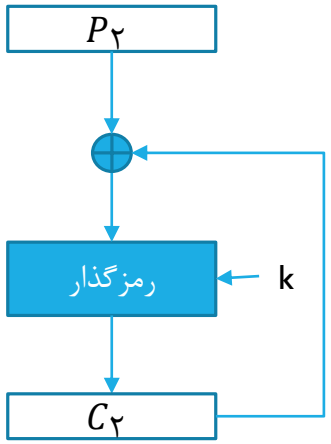
مزایای شمارنده [بنابر تحقیقات لیپمن]

- کارائی سخت‌افزاری و نرم‌افزاری
- پیش‌پردازش آسان
- سادگی
- امنیت اثبات‌پذیر
- دسترسی تصادفی

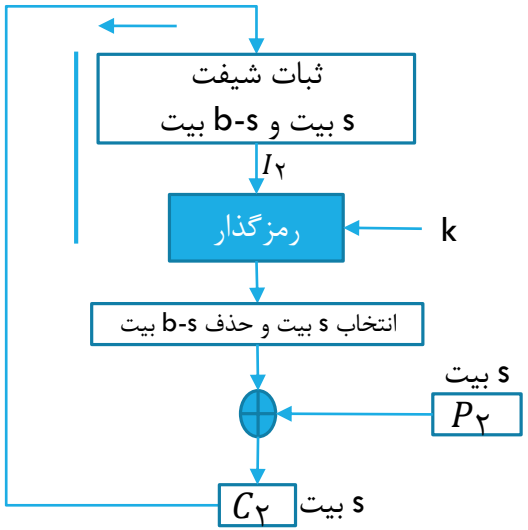
حالت شمارنده



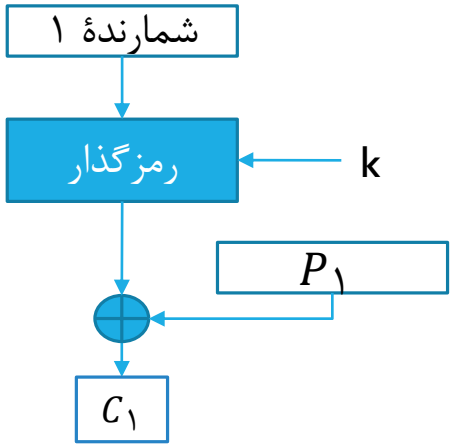
مقایسه روش‌ها



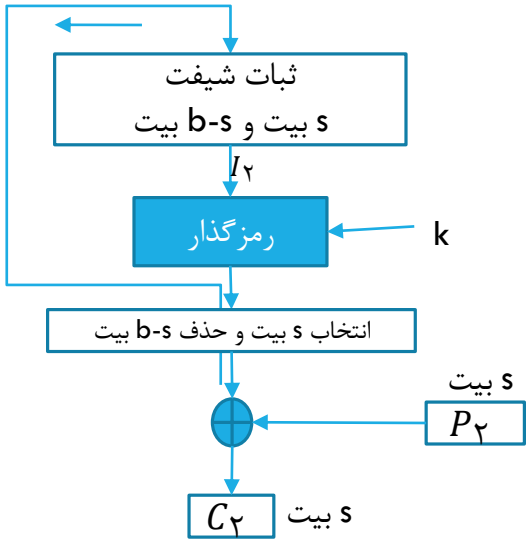
حالت زنجیر رمز



حالت بازخورد رمز



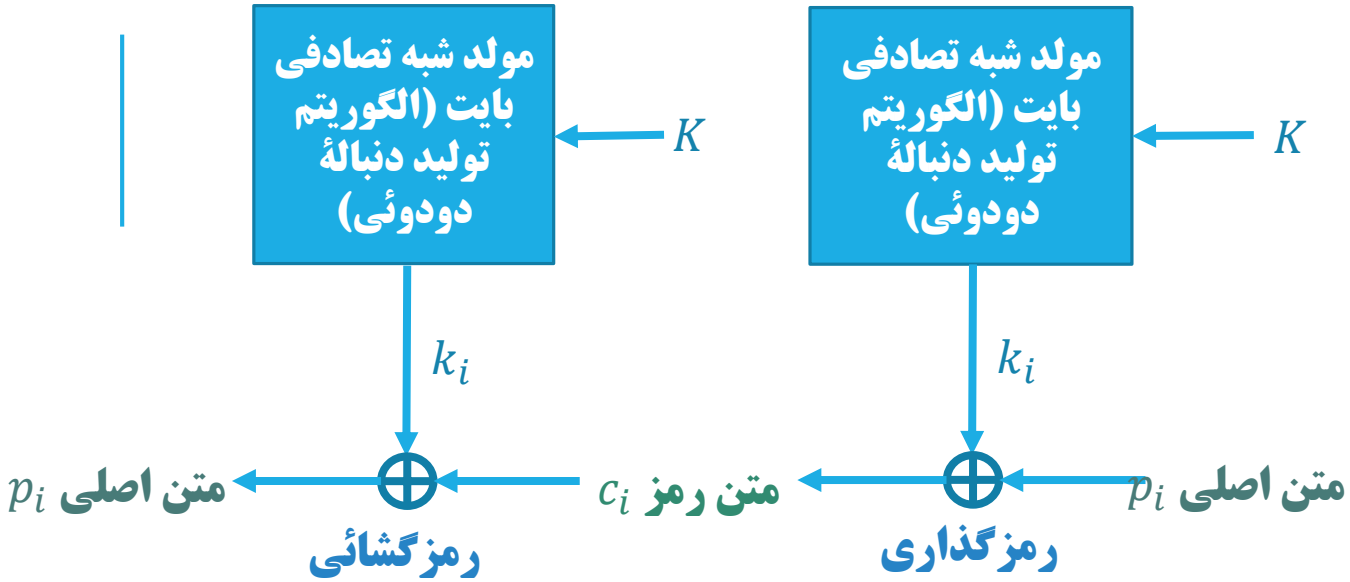
حالت شمارنده



حالت بازخورد خروجی

حالت	رمزگذاری	رمزگشائی
دفترچه متن	$C_j = E(K, P_j)$	$P_j = D(K, C_j)$
زنجیره متن رمز	$C_1 = E(K, [P_1 \oplus I])$ $C_j = E(K, [P_j \oplus C_{j-1}])$	$P_1 = D(K, C_1) \oplus I$ $P_j = D(K, C_j) \oplus C_{j-1}$
بازخورد رمز	$I_1 = I$ $I_j = LSB_{b-s}(I_{j-1}) \parallel C_{j-1}$ $O_j = E(k, I_j)$ $C_j = P_j \oplus MSB_s(O_j)$	$I_1 = I$ $I_j = LSB_{b-s}(I_{j-1}) \parallel C_{j-1}$ $O_j = E(k, I_j)$ $P_j = C_j \oplus MSB_s(O_j)$
بازخورد خروجی	$I_1 = Nonce$ $I_j = O_{j-1}$ $O_j = E(k, I_j)$ $C_j = P_j \oplus O_j$ $C_N^* = P_N^* \oplus MSB_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1}$ $O_j = E(k, I_j)$ $P_j = C_j \oplus O_j$ $P_N^* = C_N^* \oplus MSB_u(O_N)$
روش شمارنده	$C_j = P_j \oplus E(k, T_j)$ $C_N^* = P_N^* \oplus MSB_u(E(K, O_N))$	$P_j = C_j \oplus E(k, T_j)$ $P_N^* = C_N^* \oplus MSB_u(E(K, O_N))$

رمز دنباله



پردازش پیوستهٔ ورودی‌ها

در هر گام رمز یک بایت

- امکان واحدهای کوچکتر یا بزرگتر

بدون داشتن کلید ورودی پیش‌بینی‌ناپذیری دنباله شبه‌تصادفی

دنباله کلید: خروجی مولد

- یاانحصاری با تک بایت در هر گام

متن اصلی	۱۱۰۰۱۱۰۰	
دنباله کلید	01101100	⊕
متن رمز	۱۰۱۰۰۰۰۰	

متن رمز	۱۰۱۰۰۰۰۰	
دنباله کلید	01101100	⊕
متن اصلی	۱۱۰۰۱۱۰۰	

رمز دنباله

رمز دنباله شبیه لاگذاری تک‌بار
- تفاوت در نحوه تولید کلید تصادفی

موارد مهم در طراحی رمز دنباله از نظر کوما
- الف- طولانی بودن بازهٔ دنباله رمزنگاری
  - مولد شبه‌تصادفی تولیدکنندهٔ دنباله‌ای قطعی حتما تکراری
  - تعویق بیشتر زمان تکرار، سخت‌تر شدن کار تحلیل‌گر
- ب- تقریبی از اعداد تصادفی واقعی بودن دنباله کلید تا حد ممکن
- ج- بهتر است کلیدها و طول رشته‌ها طولانی باشد.
  - فعلا استفاده از ۱۲۸ بیت

سریعتر و راحت‌تر بودن روش‌های رمز دبناله

مناسب بودن جهت کاربردهایی مانند
- کانال ارتباطی داده یا پیوند تار/مرورگر که نیاز به رمزگذاری/گشائی دنباله‌ای از داده دارند

مناسب بودن رمز بلوک برای کاربردهایی که با بلوک‌های داده
- مانند انتقال فایل،ایمیل، یا پایگاه

امکان نظری استفاده از هر دو روش برای هر کاربردی

رمز دنباله با هر مولد عدد شبه‌تصافی رمزنگاری قوی می‌توان ایجاد کرد.

RC4

نوعی رمز دنباله

ران رایوست

طراحی آن جهت امنیت رسا در سال ۱۳۶۷

بر اساس تحلیل‌ها دارای تناوب $۱۰^{۱۰۰}$

انجام هشت تا شانزده عملیات ماشین برای هر بایت خروجی

استفاده در پروتکل «دسترسی حافظت‌شدهٔ بی‌سیم WPA»

- بخشی از IEE 802.11 است استفاده می‌شود.
- همچنین در SSH و کربراس گزینه‌ای اختیاری

محرمانه بودن روش مذکور برای مدت طولانی

انتشار ناشناس الگوریتم روی اینترنت در شهریور ۱۳۷۳

RC4

کلید طول-متغیری

- از ۱ تا ۲۵۶ بایت
- استفاده در مقداردهی اولیهٔ بردار حالت ۲۵۶ بایتی S
  - با اعضاء $S[0]$ $S[1]$ .... $S[255]$
- در هر زمان S دارای جایگشتی از اعداد ۸ بیتی صفر تا ۲۵۵
- تولید مقدار k از S با انتخاب روشمند یکی از ۲۵۵ مدخل جهت رمزگذاری و رمزگشائی
- سپس، اجرای دوبارهٔ جایگشت

RC4

مقداردهی اولیهٔ S

در ابتدا با $S[0] = 0$، $S[1] = ۱$، $S[0] = 0$، ...، $S[۲۵۵] = ۲۵۵$،

ایجاد بردار موقت T

استفاده از T جهت ایجاد جایگشت اولیهٔ S

- به دلیل استفاده از تابع تعویض جا⟸جایگشت تنها تاثیر ممکن

- $/*$ ***Initialization*** $*/$
- ***for*** $i = 0$ ***to*** 255 ***do***
  - $S[i] = i;$
  - $T[i] = K[i \%keylen];$

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

RC4

عدم نیاز به کلید پس از مقداردهی S

تولید دنباله

- $/*\ \boldsymbol{Stream\ Generation}\ */$
- $i, j\ =\ 0;$
- $\boldsymbol{while}\ (true)$
  - $i\ =\ (i\ +\ 1)\%256;$
  - $j\ =\ (j\ +\ S[i])\%256;$
  - $\boldsymbol{Swap}\ (S[i], S[j]);$
  - $t\ =\ (S[i]\ +\ S[j])\%256;$
  - $k\ =\ S[t];$

- جهت رمزگذاری
  - یاانحصاری k با مقدار بایت بعدی متن اصلی
- جهت رمزگشائی
  - یاانحصاری k با متن رمز

RC4

استحکام و قدرت روش

در سال

- ۱۳۸۶یافتن تهدیدی در الگوریتم تقسیم‌بندی RC4
- کاهش‌دهندهٔ زمان لازم برای یافتن کلید
- در پی آن، انجمن مهندسی اینترنت اعلام ممنوعیت استفاده از روش مذکور در TLS
- سپس، تصویب ممنوعیت استفاده از آن در کاربردهای دولتی در سازمان استانداردهٔ امریکا

منابع

[شنون]

[استالینگز]

[لاودن]