روش های نوین در سیستم‌های رمزنگاری مبتنی بر شیفت رجیسترهای با فیدبک خطی

امین جوادی نسب[1]، ابراهیم بهروزیان نژاد[2]، حامد محمدیان [3]، سجادیوسفی [4]
a.javadi62@gmail.com Behrouzian.e@gmail.com
۱،۲،۳ گروه برق و کامپیوتر دانشگاه آزاد اسلامی واحد شوشتر
٤, دانشگاه آزاداسلامی واحدعلوم تحقیقات تهران

چکیده :

رمزگذارهای جویباری یک کلاس مهمی از الگوریتمهای رمزگذاری هستند. یک رمزگذار جویباری یک الگوریتم رمزگذاری متقارن است. رمزگذارهای جویباری می‌توانند طوری طراحی شوند که با سرعتی خیلی بیشتر از هر رمزگذار بلاکی عمل کنند. رمزگذارهای بلاکی روی بلاکهای بزرگی از داده عمل می‌کنند درحالی که رمزگذارهای جویباری روی واحدهای کوچکی از پیغام که بطور معمول بیتها هستند، عمل می‌کنند. در این مقاله به بررسی دقیق ساختار درونی رمزگذارهای جویباری مبتنی برشیفت رجیسترهای با فیدبک خطی و شیفت رجیسترهای با فیدبک نقلی می‌پردازیم و مزایای این نوع رمزگذاری بر رمزگذارهای بلاکی را بیان می‌نماییم.

کلمات کلیدی

رمزگذاری - برشیفت رجیسترهای با فیدبک خطی- شیفت رجیسترهای با فیدبک نقلی

۱- مقدمه

رمزگذارهای جویباری یک کلاس مهمی از الگوریتمهای رمزگذاری هستند. آنها بطور منحصر بفردی کاراکترهای (معمولاً ارقام باینری) یک پیغام را در یک زمان با استفاده از یک انتقال رمز در زمانهای متفاوت، رمزگذاری می‌کنند. در مقابل رمزگذارهای بلاکی تمایل دارند که بطور همزمان و یک جا گروههای از کاراکترهای یک پیغام را با استفاده از یک پیغام رمزی ثابت، رمزگذاری کنند. رمزگذارهای جویباری معمولاً در سخت‌افزار سریعتر از رمزگذارهای بلاکی هستند و پیچیدگی مدارات سخت‌افزاری آنها پایین است. آنها همچنین بدلیل دارا بودن انتشار خطای محدود یا حتی بدون انتشار خطا، در موقعیتهایی که احتمال وقوع خطاهای انتقال بالا باشد، بسیار مفید واقع می‌شوند.دانش تئوری وسیعی درمورد رمزگذارهای جویباری و همچنین اصول طراحی گوناگونی برای آنها ارائه شده است. بهرجهت الگوریتمهای رمزگذاری جویباری کاملاً دلخواه در نوشته‌ها نسبتاً کم است. این موضوع تأسف بار می‌تواند تا اندازه‌ای با این حقیقت توجیه شود که بیشتر رمزگذارهای جویباری به صورت اختصاصی و محرمانه مورد استفاده قرار می‌گیرند. در مقابل پیشنهادهای متعددی چه به صورت استاندارد و چه به صورت کلی درمورد رمزگذارهای بلاکی ارائه می‌شود. با

وجود این، رمزگذارهای جویباری بدلیل مزایای مهمشان، امروزه بطور وسیعی مورد استفاده قرار می‌گیرند. رمزگذارهای جویباری می‌توانند یا کلید متقارن باشد یا کلید عمومی. تمرکز این مقاله بر روی رمزگذارهای جویباری کلید متقارن است. رمزگذارهای بلاکی پیغام را با بلاکهای نسبتاً بزرگی پردازش می‌کنند (n>64 bits). در این نوع رمزگذارها از تابعی یکسان برای رمزگذاری بلاکها استفاده می‌شود. بنابراین رمزگذارهای بلاکی بدون حافظه هستند. در مقابل، رمزگذارهای جویباری پیغام را در بلاکهایی به کوچکی یک تک بیت پردازش می‌کنند و تابع رمزگذاری ممکن است تغییر کند. بنابراین رمزگذارهای جویباری دارای حافظه هستند. در بعضی مواقع به آنها رمزگذارهای حالت نیز گفته می‌شود زیرا رمزگذار نه تنها به کلید و پیغام وابسته است بلکه به حالات جاری نیز وابسته است. این اختلاف میان رمزگذارهای جویباری و رمزگذارهای بلاکی قطعی نیست یعنی با اضافه کردن یک مقدار حافظه کم به رمزگذارهای بلاکی، رمزگذارهای جویباریی با بلاکهای بزرگ نتیجه می‌شود.

.

۲ –مولدهای دنباله‌های شبه اتفاقی و رمزگذاری جویباری

۲–۱ مولدهای متجانس خطی

مولدهای متجانس خطی نوعی از مولدهای شبه اتفاقی هستند که فرم کلی آنها بصورت زیر می‌باشد.

$$X_n = (aX_{n-1} + b) \bmod m$$

که در رابطه بالا، X_n نشانگر عدد n ام در دنباله، X_{n-1} بیانگر عدد قبلی آن می‌باشد. متغیرهای m,b,a ثابت هستند که a را عامل ضرب کننده، b را عامل افزاینده و m را قدرمطلق این مولدها می‌نامند. هسته و مقدار اولیه این مولدها X_0 می‌باشد. حداکثر پریود این مولد برابر با m خواهد بود. بصورتی که اگر m,b,a بطور صحیح انتخاب شوند مولدی با حداکثر پریود که مولدی با حداکثر طول هم نامیده می‌شود خواهیم داشت. در واقع می‌توان با انتخاب صحیح m,b,a یک مولد با پریود m دست پیدا کرد. به عنوان مثال در انتخاب b,a باید دقت کرد که b باید نسبت به m اول باشد.

از مهمترین ویژگیهای مولدهای متجانس خطی می‌توان به سرعت بالای آنها و کم‌بودن عملیات لازم برای تولید هر بیت اشاره کرد.

متأسفانه مولدهای متجانس خطی کاربردی در رمزنگاری ندارند و برای کاربردهای رمزنگاری از آنها استفاده نمی‌شود زیرا دنباله‌های بدست آمده قابل پیش بینی هستند. اولین بار این مولدها توسط شخصی به نام Jim Reeds شکسته شد. پس از وی نیز Joan Boyar موفق به شکستن آنها شد. حتی توانستند مولدهای درجه دوم که فرم کلی آنها بصورت زیر می‌باشد را نیز بشکنند.

$$X_n = (aX_{n-1}^2 + bX_{n-1} + c) \bmod m$$

فعالیت آنها در شکستن مولدها به اینجا ختم نشد. و توانستند مولدهای درجه سوم به فرم کلی زیر را نیز از پای درآورند.

$$X_n = (aX_{n-1}^3 + bX_{n-1}^2 + cX_{n-1} + d) \bmod m$$

پس به نظر می‌رسد که مولدهای چند جمله‌ای دیر یا زود شکسته می‌شوند پس امن نیستند. پس نمی‌توان به مولدی با پارامترهای نامشخص (m,b,a) اطمینان داشت.

ولی با این حال هنوز هم مولدهای متجانس خطی برای کاربردهای غیر رمزنگاری مفید بنظر می‌رسند که از کاربردهای آنها می‌توان به شبیه‌سازها اشاره کرد. این مولدها در آزمایشهای تجربی کاملاً مؤثر و مفید هستند و رفتارهای آماری خوبی را از خود نشان می‌دهند. پس دلیل اینکه این مولدها هنوز هم پیاده‌سازی می‌شوند اینست که اطلاعات قابل توجهی را می‌توان از کارکردن با آنها بدست آورد.

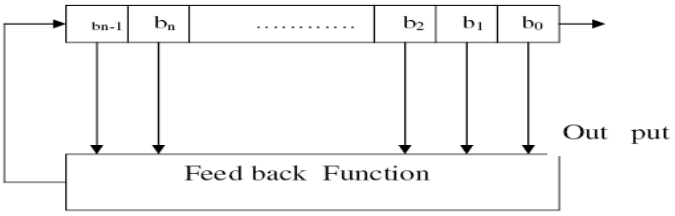
۲–۲) ترکیب مولدهای متجانس خطی

یکی از راههای بهینه کردن مولدهای متجانس خطی، استفاده ترکیبی از چندین مولد می‌باشد. اگرچه ترکیب مولدها باعث نشده که برای کاربردهای رمزنگاری امن شوند اما بوجود آوردن دنباله‌هایی با پریود طولانی‌تر و بدست آوردن اطلاعات بیشتر در بعضی آزمایشها، باعث شده است که فرآیند ترکیب مولدهای متجانس خطی مطرح شود.با توجه به مشخصه‌هایی که از این مولدها ذکر شد. نتیجه گرفتیم که این مولدها ناامن هستند و برای کاربردهای رمزنگاری کاربرد ندارند. پس نوع دیگری از مولدها ارائه شده است که بر مبنای شیفت رجیسترهای فیدبک خطی می‌باشند.

۳) شیفت رجیسترهای فیدبک خطی

دنباله‌های حاصل از شیفت رجیسترها در رمزنگاری و هم در تئوری کدینگ کاربرد دارند. مبنای کاری رمزگذارهای جویباری نیز براساس شیفت رجیسترها می‌باشند که بیشتر برای رمزنگاری کاربردهای نظامی استفاده می‌شوند. شیفت رجیسترهای فیدبک خطی[1] (LFSR)در پیاده‌سازی سخت‌افزاری نیز بسیار راحت و ساده می‌باشند. و همچنین دارای خواص آماری خوبی می‌باشند.

اجزاء تشکیل دهنده شیفت رجیستر فیدبک‌دار، تابع فیدبک و شیفت رجیستر می‌باشند. شیفت رجیستر عبارتست از تعدادی عنصر تأخیری که در کنار هم چیده شده‌اند و هر کدام قابلیت ذخیره‌سازی یک بیت را دارد. و دارای یک ورودی و یک خروجی می‌باشد. طول یک شیفت رجیستر برحسب bit حساب می‌شود و شامل تعداد بیت‌هایی است که می‌توانند در شیفت رجیستر قرار بگیرند. اگر یک شیفت رجیستر n بیتی نامیده شود به این معناست که n عنصر تأخیری (۰و۱و۲و... و ۱ـ n) کنار هم قرار گرفته‌اند. با وجود n بیت در داخل شیفت رجیسترها، در هر واحد زمانی یک بیت به عنوان خروجی درنظر گرفته می‌شود که همان محتوای خانه صفرام شیفت رجیستر می‌باشد. پس با هر کلاک خانه صفرام به عنوان خروجی و بخشی از دنباله خروجی درنظر گرفته می‌شود. و همه بیتها یک بیت به راست شیفت خواهند داشت. در واقع محتوای حالت I به حالت ۱ـ I منتقل خواهد شد. به این ترتیب حالت ۱ـ n فاقد مقدار می‌باشد که برای این حالت نیز با استفاده از تابع فیدبک مقداری بدست آمده و در آن قرار می‌گیرد. شکل کلی یک شیفت رجیستر فیدبک‌دار در شکل (۱) دیده می‌شود.

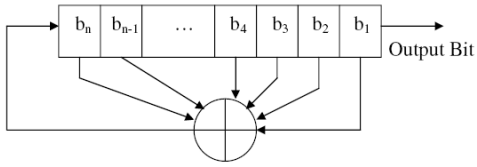


شکل (۱) : شیفت رجیستر فید بک دار

پریود یک شیفت رجیستر عبارتست از طول دنباله خروجی قبل از اینکه دنباله شروع به تکرار شدن کند. یکی از مهمترین عواملی که، رمزگذاری جویباری بر مبنای شیفت رجیسترها می‌باشد اینست که آنها از نظر پیاده‌سازی سخت‌افزاری بسیار آسان هستند و می‌توان تجزیه و تحلیل آنها را با استفاده از تکنیکهای ریاضی به راحتی انجام داد. تابع فیدبک نیز عبارتست از تابعی که بر روی بیتهای مشخصی از شیفت رجیستر عمل می‌کند و مقدار حالت ۱ـ nرا تولید می‌کند. به بیتهایی که در تابع فیدبک استفاده می‌شوند دنباله Tap گفته می‌شود. البته درحالت ساده‌تر این تابع همان XOR درنظر گرفته می‌شود.تابع فیدبک را می‌توان بصورت زیر درنظر گرفت.

$$F(b_0,b_1,\ldots,b_{n-1}) = c_0b_0 + c_1b_1 + \ldots c_{n-1}b_{n-1}$$

ثابتهای $C_{n-1},\ldots ,C_1,C_0$ ضرایب فیدبک نامیده می‌شوند. و به عنوان سوئیچ درنظر گرفته می‌شوند به طوری که اگر C_i =۱ باشد سوئیچ بسته است درغیر این صورت سوئیچ باز است. در شکل (۲)شیفت رجیستر فیدبک خطی دیده می‌شود. LFSR ها پرکاربردترین نوع شیفت رجیسترها هستند که در رمزنگاری استفاده می‌شود.



شکل (۲)شیفت رجیستر فید بک دار خطی

[1](Linear Feedback Shift Register)

یک شیفت رجیستر فیدبک خطی به طول n، حداکثر ۱ـ۲^n حالت متفاوت می‌تواند داشته باشد. این بدان معنی است که یک شیفت رجیستر فیدبک خطی به طول n، حداکثر می‌تواند دنباله‌ای شبه اتفاقی به طول ۱ـ۲^n بیت را قبل از اینکه خروجی تکرار شود تولید کند. شاید برای شما سؤال پیش بیاید که چرا ۲^n حالت نمی‌توان تولید کرد و عدد "1" از آن کم شده است؟ به این دلیل که در شیفت رجیسترهای فیدبک خطی حالت تمام صفر نداریم چون اگر به هر دلیلی مقدار تمام صفر در شیفت رجیستر قرار بگیرد، دنباله خروجی رشته‌ای از صفرها را تولید خواهد کرد. و از حالت تمام صفر خارج نخواهد شد که این حالت اصلاً مفید نخواهد بود. پس شیفت رجیسترهایی با دنباله Tap معین که طولی برابر با ۱ـ۲^n دارند را شیفت رجیستر با طول ماکزیمم می‌نامیم و آنرا m-Sequence می‌نامند.

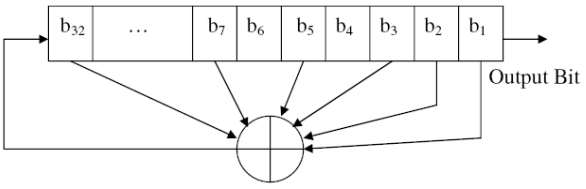
برای اینکه حالت تمام صفر نداشته باشیم باید از یک مدار اضافی استفاده شود تا درصورتی که حالت تمام صفر در شیفت رجیستر بوجود آمد آنرا تشخیص دهد و آنرا به حالت خاصی ریست کند. معمولاً برای اینکار از گیت NOR استفاده می‌کنند.برای یک شیفت رجیستر فیدبک خطی با طول ماکزیمم یک چندجمله‌ای F(x) وجود دارد که این چند جمله‌ای از دنباله Tap بعلاوه مقدار ثابت "1" بدست می‌آید.

$$F(x) = \text{Tap Sequence} + 1$$

این چند جمله‌ای F(x) باید Primitive به پیمانه ۲ باشد. درجه این چند جمله‌ای برابر طول شیفت رجیستر خواهد بود. چند جمله‌ای Primitive از درجه n، چندجمله‌ای است که غیر قابل تجزیه‌شدن باشد. به عبارتی Primitive قابل تقسیم بر $X^{2^n-1}+1$ خواهد بود ولی قابل تقسیم بر X^d+1 نخواهد بود که d قابل تقسیم بر 2^n-1 باشد. درحالت کلی راه‌حل ساده‌ای برای تولید چندجمله‌ای‌های Primitive به پیمانه ۲ با یک درجه مشخص وجود ندارد. ساده‌ترین راهی که به نظر می‌رسد اینست که چندجمله‌ای‌ها را به صورت تصادفی انتخاب کنیم و بعد Primitive بودن یا نبودن آنها را تست کنیم. که این عملیات خیلی پیچیده‌ای را می‌طلبد اما برای اینکار بسته‌های نرم‌افزاری ریاضی وجود دارد که این عملیات را انجام می‌دهند عبارتی مثل [۳۲و۷و۶و۵و۳و۲و۱و۰] وجود دارد که این عبارت بدان معناست که چند جمله‌ای زیر Primitive می‌باشد.

$$X^{32}+X^7+X^5+X^{3+}X^2+X+1$$

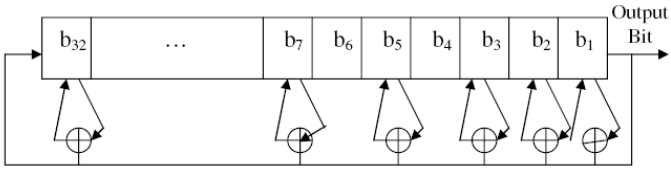
تبدیل چندجمله‌ای فوق به یک شیفت رجیستر فیدبک خطی با پریود ماکزیمم کار بسیار ساده‌ای می‌باشد. اولین عدد نشانگر طول شیفت رجیستر می‌باشد و آخرین عدد همیشه صفر است که می‌توان آنرا نادیده گرفت کلیه اعداد بجز عدد صفر نشانگر دنباله Tap می‌باشند. پس به این ترتیب مثال فوق بیان کننده یک شیفت رجیستر فیدبک خطی به طول ۳۲ بیت می‌باشد که بیت جدید آن از XOR کردن بیتهای ۳۲و۷و۵و۳و۲و۱ با هم بوجود می‌آید. شکل(۳) این شیفت رجیستر فیدبک خطی را نشان می‌دهد.



شکل(۳) LFSRخطی ماکزیمم به طول ۳۲ بیت

این LFSR دارای پریود ۱ـ$۲^{۳۲}$ می‌باشد که همان پریود ماکزیمم می‌باشد . به عنوان مثال یک LFSR به طول ۹۶۸۹ بیت دارای سه ضریب می‌باشد ([۰،۸۴،۹۶۸۹]) به این علت به چندجمله‌ای‌ها، چند جمله‌ای‌های پراکنده گفته می‌شود. همین عامل پراکندگی در چندجمله‌ای‌ها، باعث ضعیف شدن آنها خواهد شد. و به همین ترتیب باعث شکسته شدن الگوریتم نیز خواهد شد. پس راه‌حل بهتر اینست که از چندجمله‌اهایی با ضرایب بیشتر استفاده شود تا امنیت نیز افزایش یابد. و برای کاربردهای رمزنگاری مناسب شود. به این چندجمله‌ای‌ها که ضرایب آنها بیشتر است چندجمله‌ای‌های متراکم گفته می‌شود. به هر حال اگر از چندجمله‌ای‌های متراکم برای کاربردهای خاص، مخصوصاً بعنوان کلید رمزنگاری استفاده شود برای استفاده مؤثر، بهتر است از LFSR هایی با طول کمتر استفاده شود. تولید چندجمله‌ای‌های Primitive به پیمانه ۲ که متراکم باشند آسان نیست. ولی درحالت کلی تولید چندجمله‌ای‌های Primitive که دارای درجه k باشد. دانستن فاکتورگیری و عامل‌بندی ۱ـ۲^k ضروری به نظر می‌رسد. LFSR ها نوعی از مولدهای دنباله‌های شبه اتفاقی ایده‌ال هستند که دارای ویژگیهای خیلی مناسبی هستند ولی با این حال بعضی ویژگیهای غیراتفاقی هنوز هم آزاردهنده است. بیتهای متوالی دارای ویژگی خطی می‌باشند که همین عامل باعث می‌شود که استفاده کمی در کاربردهای رمزنگاری داشته باشند. برای یک LFSR به طول n,n بیت خروجی بعدی به عنوان حالت داخلی درنظر گرفته می‌شود. حتی اگر نحوه

فیدبک و تابع فیدبک یک LFSR مشخص نباشد. اگر بتوان ۲n بیت خروجی مولد را کشف کرد با استفاده از الگوریتم Berlekamp – Massey می‌توان تابع فیدبک را بدست آورد. بنابراین اعداد بزرگی که بصورت تصادفی توسط بیتهای متوالی یک رشته تولید می‌شوند بشدت همبسته هستند و برای نوع‌های معین در کاربردهای خاص اصلاً تصادفی و اتفاقی نیستند. حتی LFSR ها به عنوان بلاکهای سازنده در الگوریتم‌های رمزنگاری مورد استفاده قرار می‌گیرد. عوض اینکه بیتهای موجود در دنباله Tap برای تولید چپ‌ترین بیت مورد استفاده قرار گیرند بلکه هر بیت با بیت خروجی XOR می‌شود و در موقعیت بیتی خودش جای می‌گیرد و کم ارزشترین بیت نیز به عنوان خروجی درنظر گرفته می‌شود. به این نوع مولد، پیکربندی Galois گفته می‌شود که در شکل (٤) فرم کلی این مولد دیده می‌شود.



شکل(٤) Galois LFSR

مولد بالا بصورت تک عملگری می‌باشد (عملگر XOR) پس می‌تواند بصورت موازی هم این عملگرها انجام شوند و چندجمله‌ای‌های فیدبک متفاوت نتایج متفاوتی را خواهند داشت. پیکربندی Galois در پیاده‌سازی سخت‌افزاری بخصوص در مدارات VLSI اختصاصی ، دارای سرعت بیشتری می‌باشد.پس ایده اصلی اینست که اگر از سخت‌افزار استفاده می‌کنید که دارای رفتار خوبی در شیفت‌ها می‌باشد بهتر است از پیکربندی فیبوناچی استفاده شود ولی اگر از ویژگی موازی عملیات بخواهید استفاده کنید از پیکربندی Galois استفاده کنید.

٤)رمزگذارهای جویباری مبتنی بر LFSR ها

اصلی‌ترین روش برای طراحی مولد جریان کلید استفاده از LFSR ها می‌باشد که بسیار روش ساده‌ای هم است. برای این کار باید چندین LFSR که طولهای متفاوت و چندجمله‌ای‌های فیدبک متفاوتی دارند را انتخاب کنیم (اگر طولها نسبت به هم اول باشند و چندجمله‌ای‌های فیدبک نیز اول باشند، هم مولدها دارای طول ماکزیمم خواهند بود) کلید این روش، حالت ابتدایی LFSR ها می‌باشد.

در هر واحد زمانی یک بیت به عنوان خروجی گرفته می‌شود که حاصل شیفت LFSR ها می‌باشد که عمل شیفت LFSR ها در بعضی مواقع کلاکینگ هم گفته می‌شود و بیت خروجی نهایی نیز توسط تابعی که احتمالاً غیرخطی هم است بدست می‌آید در واقع این تابع بر روی تعدادی از بیتهای LFSR ها اعمال می‌شود. و بیت خروجی نهایی را بوجود می‌آورد به این تابع، تابع ترکیب و به مولد حاصل، مولد ترکیبی گفته می‌شود. اگر بیت خروجی، تابعی از یک LFSR باشد به مولد حاصل مولد فیلتر گفته می‌شود. کنترل کلاکی می‌تواند بصورت پیش خورد انجام شود یعنی خروجی یک LFSR، کلاک LFSR های سطوح بعدی را کنترل کند یا اینکه بصورت بازخورد باشد، که دراین صورت خروجی LFSR کلاک خودش را کنترل می‌کند. اگرچه این مولدها ساخته شده‌اند و مورد استفاده هم قرار می‌گیردن اما دست کم احتمالاً این مولدها در حملات همبستگی آسیب‌پذیر هستند. ولی درحالت کلی این مولدها برای کاربردهای کنونی امن هستند. چه بسا در آینده‌ای نزدیک کاملاً منسوخ شوند. خیلی از اشخاصی که صاحب نظریه در مسائل رمزنگاری هستند بر این عقیده‌اند که رمزنگاری عبارتست از ترکیبی از ریاضیات محض و عامل دیگری بنام درهم برهم شدن و بدون درنظر گرفتن عامل درهم برهم شدن رمزنگاری، کارایی خوبی نخواهد داشت. پس دانش ریاضیات باعث آشنایی بیشتر با ساختار LFSR ها خواهد شد و LFSR هایی را ارائه می‌دهد که طول ماکزیمم داشته باشند و مخلوط LFSR حاصل با عوامل درهم برهم کنی غیرخطی پیچیده شده، باعث شده که بدست آوردن ترکیب رجیستر برای حمله کننده‌ها سخت باشد و شکستن آن نیز مشکل باشد. همین مزیت گفته شده باعث می‌شود که در بیشتر مواقع رمزگذارهای جویباری به الگوریتم‌های بلاکی ترجیح داده شوند.در ادامه با یکسری از مولدهای جریان کلید مبتنی بر LFSR بیشتر آشنا می‌شویم اما اطلاعات کافی درمورد اینکه کدام یک از این مولدها در دنیای واقعی مورد استفاده قرار می‌گیرند در دسترس نیست در ضمن بیشتر تکنیکهایی که گفته می‌شود فقط جنبه تئوری داشته و بعضی از آنها نیز شکسته شده‌اند شاید تنها جزئی از تکنیکهایی

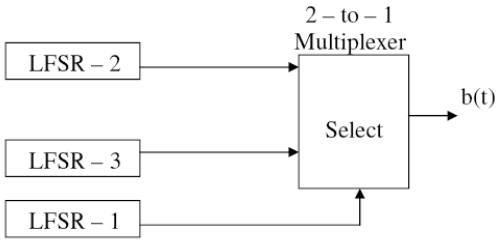
که گفته خواهد شد هنوز هم امن باشند. از آنجایی که رمزگذارهای مبتنی بر LFSR ها به طور عمومی پیاده‌سازی سخت‌افزاری می‌شوند، در شکلهای کشیده شده نیز از سمبل‌های منطق الکترونیکی استفاده شده است.

٤-١)مولد Geffe

این مولد جریان کلید. از سه عدد LFSR استفاده می‌کند. این سه LFSR بصورت غیرخطی ترکیب می‌شوند. دو تا از LFSR ها به عنوان ورودی مالتی پلکسر و LFSR سومی به عنوان کنترل کننده خروجی مالتی پلکسر مورد استفاده قرار می‌گیرند. اگر a_3,a_2,a_1 خروجی‌های LFSR باشند خروجی مولد Geffe توسط تابع زیر محاسبه می‌شود.

$$B = (a_1 \text{ and } a_2) \text{ XOR } ((\text{not } a_1) \text{ and } a_3)$$

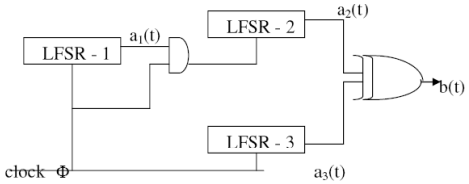
شکل کلی این مولد در شکل (٥) دیده می شود. اگر طولهای LFSR ها به ترتیب n_3,n_2,n_1 باشند پیچیدگی خطی مولد برابر (n_1+1) $n_2+n_1n_3$ خواهد شد. پریود این مولد برابر با کوچکترین مضرب مشترک پریودهای سه مولد خواهد بود. با فرض اینکه درجه چندجمله‌ای‌های فیدبک نسبت به هم اول باشند پریود کل مولد برابر با حاصلضرب پریود سه LFSR می‌باشد. اگرچه این مولد به ظاهر خوب به نظر می‌رسد ولی در کاربردهای رمزنگاری بسیار ضعیف می‌باشد و در حملات همبستگی مقاوم نمی‌باشد. خروجی مولد در ٧٥ درصد زمانها برابر با خروجی LFSR 2 می‌باشد. بنابراین اگر Tap های فیدبک را به طریقی بتوان بدست آورد. می‌توان مقدار اولیه‌ای که برای LFSR 2 درنظر گرفته شده است را حدس زد و ترتیب دنباله خروجی آن رجیستر را بدست آورد. پس تعداد دفعاتی که خروجی LFSR 2 با خروجی مولد برابر است را شمرد اما اگر مقدار اولیه‌ای که برای LFSR 2 درنظر گرفته‌ایم اشتباه باشد دو دنباله حاصل در ٥٠ درصد زمانها با هم یکسان خواهد بود. و اگر مقدار درنظر گرفته شده درست باشد دو دنباله در ٧٥ درصد حالتها با هم یکسان خواهند بود. و به طور مشابه خروجی مولد در ٧٥ درصد زمانها برابر با خروجی LFSR 3 خواهد بود پس با وابستگی‌های گفته شده مولدهای جریان کلید به روش فوق زود شکسته خواهند شد. به عنوان مثال اگر چند جمله‌ای‌های Primitive فقط از سه ضریب تشکیل شده باشند و طول بزرگترین LFSR در این مولد برابر با n باشد با داشتن ٣٧n بیت از دنباله خروجی کافی است تا تمام حالتهای داخلی سه LFSR را بازسازی کرد.



شکل (٥) مولد Geffe

٤-٢)مولد Beth – Piper Stop – and - Go

این مولد در شکل (٦) نشان داده شده است. در این مولد از خروجی یک LFSR برای کنترل کلاک مربوط به LFSR بعدی استفاده شده است. درشکل بالا کلاک LFSR -2 توسط خروجی LFSR -1 کنترل می‌شود بنابراین LFSR –2 درصورتی در زمان t تغییر خواهد داشت به شرطی که LFSR –1 در زمان t ـ١ تغییر داشته باشد. این مولد نیز در مقابل حملات همبستگی امن نیست. البته لازم به اشاره است که هنوز پیچیدگی خطی این مولد حساب نشده است.

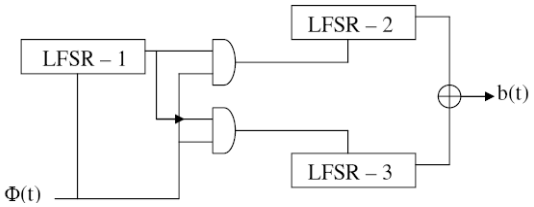


شکل (٦) مولد Beth – Piper Stop – and – Go

٤-٣) مولد Alternating Stop – and - Go

این مولد از سه عدد LFSR با طولهای متفاوت تشکیل شده است. این مولد در شکل (۷) نشان داده شده است. همانطور که در شکل دیده می‌شود LFSR –2 زمانی که خروجی LFSR -1 برابر با "1" شود کلاک خواهد خورد و به همین ترتیب اگر خروجی LFSR -1 برابر با صفر باشد LFSR -3 کلاک خواهد خورد. این مولد پریود طولانی و همچنین پیچیدگی خطی بالایی داراست. اگرچه درباره این مولد، حمله همبستگی برعلیه LFSR -1 مطرح است ولی اساساً این عامل نمی‌تواند باعث ضعیف شدن مولد شود.

درضمن خروجی نهایی از XOR کردن خروجیهای LFSR –2 و LFSR -3 بدست می‌آید.



شکل(۷) مولد Alternating Stop – and – Go

٤-٥)مولد Threshold

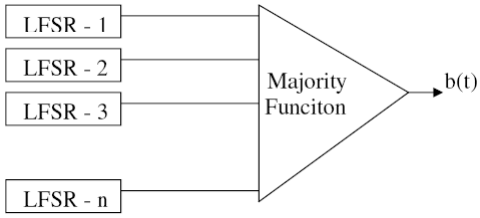
این مولد برای اینکه بتواند تا حدی بر مشکلات امنیتی موجود در مولدهای قبلی غلبه کند از تعداد متغیرهای LFSR ها استفاده می‌کند. تئوری مطرح شده در این زمینه اینست که اگر از تعداد بیشتری از LFSR ها استفاده شود شکستن رمزگذار به مراتب سخت‌تر خواهد شد. این مولد در شکل(۸) نشان داده شده است. همانطور که درشکل دیده می شود خروجی تعداد زیادی از LFSR ها (که در بیشتر مواقع این تعداد فرد می‌باشد) به عنوان ورودی تابع اکثریت درنظر گرفته شده است و اگر طول همه LFSR ها نسبت به هم اول باشند و همه چندجمله‌ای‌های فیدبک Primitive باشند حداکثر پریود را خواهیم داشت. عملکرد این مولد به این صورت است که از ورودیها، اکثریت را انتخاب می‌کند بطوری که اگر بیش از نصف ورودیها مقدار "۱" داشته باشد خرجی مولد نیز "۱" خواهد شد و برعکس اگر بیش از نصف ورودیها مقدار "0" داشته باشد خروجی مولد نیز "0" خواهد شد.با فرض اینکه مولد فوق با سه عدد LFSR ساخته شده باشد تابع خروجی بصورت زیر خواهد شد.

$$B = (a_1 \text{ and } a_2) \text{ XOR } (a_1 \text{ and } a_3) \text{ XOR } (a_2 \text{ XOR } a_3)$$

این مولد تا حد زیادی شبیه مولد Geffe می‌باشد. با این تفاوت که پیچیدگی خطی بالایی دارد. پیچیدگی خطی این مولد با شرط اینکه طول سه LFSR به ترتیب n_3,n_2,n_1 باشد بصورت زیر می‌باشد.

$$n_1n_2 + n_1n_2 + n_2n_3$$

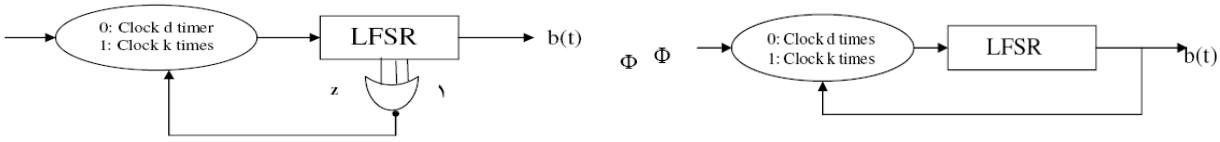
این مولد زیاد هم امن نیست. زیرا هر بیت خروجی مولد، اطلاعات خیلی خوبی را درمورد حالتهای LFSR ها نشان می‌دهد که باعث می‌شود در مقابل حملات همبستگی آسیب‌پذیر باشد. به دلیل ضعفهای گفته شده زیاد توصیه نمی‌شود که استفاده شود.



شکل (۸) مولد Threshold

٤-٦)مولدهای Self – Decimated

مولدهای Self – Decimated مولدهایی هستند که کلاکشان را خودشان کنترل می‌کنند. این نوع مولدها در دو نـوع مختلـف ارائـه شـده اسـت. اولین پیشنهاد توسط Rainer Rueppel ارائه شده است که در شکل (۹) الف نشان داده شده است. دومین پیشنهاد توسط Dieter Gollmann Bill Clambers , ارائه شده است که در شکل (۹)ب نشان داده شده است.

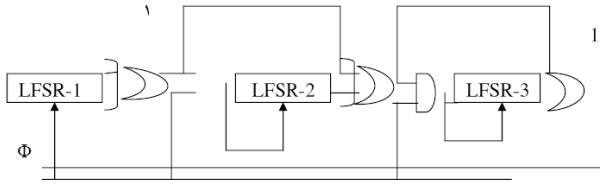


شکل (۹) ب　　شکل (۹) الف

درمولد Rueppel اگر خروجی LFSR، "0" باشد LFSR به اندازهٔ d دفعه کلاک خواهد خورد ولی اگر خروجی آن "1" باشد LFSR به اندازه k دفعه کلاک می‌خورد. مولد Gollmann پیچیده‌تر از مولد Rueppel می‌باشد ولی ایده اصلی آنها یکی است. متأسفانه هر دو مدل پیشنهاد شده امن نیستند اگرچه بعضی پیشنهادها برای بهبود بخشیدن به مشکلات کنونی ارائه شده ولی هیچکدام تأثیر زیادی در امنیت مولد نداشته است.

٤-٧مولد Gollman cascade

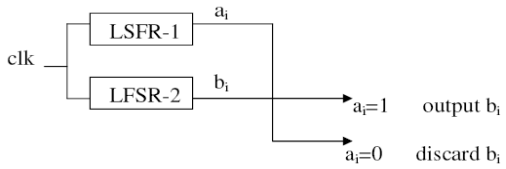
این مولد در شکل (۱۰) نشان داده شده است. این مولد یک نوع خیلی قوی از نوع مولدهای Stop – and – Go می‌باشد. این مولد از یک سـری LFSR تشکیل شده است که کلاک هر LFSR توسط LFSR قبلی کنترل می‌شود. اگـر خروجـی LFSR - 1 در زمـان ۱ـ t برابـر "1" باشـد LFSR –2 کلاک می‌خورد. و اگر خروجی LFSR –2 در زمان ۱ـ t برابر "1" باشد LFSR –3 کلاک می‌خورد و به همین ترتیب ادامه می‌یابد. خروجی مولد نیز خروجی آخرین LFSR می‌باشد. اگر همه LFSR ها دارای طول مشابهی برابر با n باشند و سیستم از k عـدد LFSR تـشکیل شده باشد پیچیدگی خطی سیستم برابر $n(2^n-1)^{k-1}$ خواهد بود.ایده آبشاری یک ایده سرد می‌باشد. بنظر می‌رسد که ایـن روش یـک روش خیلـی ساده‌ای است که می‌توان با آن دنباله‌هایی با پریودهای بالا تولید کرد و پیچیدگی خطی بزرگی خواهد داشت. درضمن ویژگیهای آماری خـوبی را از خود نشان می‌دهد. این روش درمقابل حمله‌ای به نام Lock – in آسیب‌پذیر می‌باشد. حمله Lock – in تکنیکی استفاده می‌کند که حملـه کننـده سعی می‌کند که ورودی آخرین شیفت رجیستر در آبشار را بازسازی کند سپس این عمل را برای رجیستر قبلی تکرار می‌کند در واقع رمزشکننده بـا شکستن رجیستر به رجیستر، کل مولد را می‌شکند.



شکل (۱۰) مولد Gollman cascade

٤-٨ مولد Shrinking

کنترل کلاک این مولد نسبت به مولدهای قبلی کاملاً متفاوت می‌باشد. فرض کنید مولدی با دو LFSR ساخته شده است کـه کـلاک هـم بـرای دو LFSR وارد می‌شود اگر خروجی LFSR - 1، "۱" باشد در این صورت خروجی مولد همان خروجی LFSR – 2 خواهد بـود و اگـر خروجـی LFSR – 1، "0" باشد خروجی دو LFSR بیرون ریخته می‌شود سپس باز مولد کلاک خورده و به همین ترتیب ادامه پیدا می‌کنـد. ایـن مولـد در شکل (۱۱) نشان داده شده است.

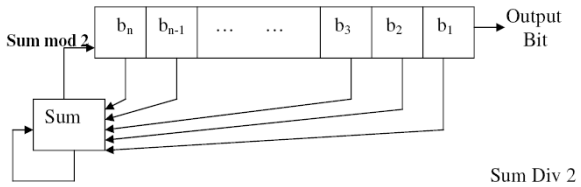


شکل (۱۱) مولد Shrinking

ایده این روش ساده می‌باشد و انصافاً هم کارا و تاحدی هم امن می‌باشد. اگر چندجمله‌ای‌های فیدبک پراکنده باشند، مولد آسیب‌پذیر خواهد بود. ولی مشکل غیر از این مسأله متوجه این مولد نیست حتی این مولد امروزه در بیشتر کاربردها مورد استفاده قرار می‌گیرد تنها مشکل پیاده‌سازی این روش اینست که نرخ خروجی این مولد منظم نیست به این ترتیب که اگر $LFSR - 1$ دارای تعداد زیادی صفر در کنار هم داشته باشد خروجی مولد نیز چیزی نخواهد بود ولی برای حل مسأله فوق پیشنهاد می‌شود که از بافرکردن خروجی‌ها استفاده شود.

۵) شیفت رجیسترهایی با فیدبک نقلی (FCSR)

شیفت رجیسترها با فیدبک نقلی ازنظر ساختاری بسیار شبیه به LFSR می‌باشد. هر دوی آنها از شیفت رجیستر و تابع فیدبک ساخته شده‌اند. فقط تفاوت آنها در این است که FCSR دارای یک رجیستر کری می‌باشد. ساختار کلی این نوع شیفت رجیسترها در شکل (۱۲) نشان داده شده است. عوض اینکه بیتهای موجود در دنباله Tap با هم XOR شوند در FCSR این بیتها با هم جمع شده و حاصل نیز با محتوای رجیستر کری جمع زده می‌شود. mod 2، نتیجه حاصل، حساب شده و به عنوان بیت جدید بشمار می‌آید درضمن div 2، نتیجه حاصل نیز مقدار جدیدی است که در رجیستر نقلی قرار می گیرد.



شکل(۱۲) شیفت رجیستر با فیدبک نقلی

چندین نکته در مورد این شیفت رجیسترها وجود دارد اولاً اینکه رجیستر کری یک بیت تنها نیست بلکه یک عدد است. اندازه رجیستر کری باید حداقل Log_2t باشد که t نشان دهنده تعداد ضرایب موجود در چندجمله‌ای می‌باشد. یا به عبارت دیگر تعداد خانه‌هایی که در عمل جمع تأثیر گذارند.ثانیاً: قبل از اینکه FCSR شروع به تکرار کردن پریود کند یک حالت ناپایدار آغازین وجود دارد. ثالثاً: اگر n طول شیفت رجیستر باشد حداکثر پریود FCSR برابر ۲^nـ۱ نخواهد بود بلکه ماکزیمم پریود برابر q ـ۱ خواهد بود که q را connection integer می‌نامند. و بصورت زیر حساب می‌شود.

$$Q = 2q_1 + 2^2q_2 + 2^4q_4 + \ldots + 2^nq_n - 1$$

لازم به توضیح است که q_i ها از چپ به راست شماره‌گذاری می‌شوند. از آنجایی که حالت اولیه در FCSR وابسته به کلید رمزگذار جویباری می‌باشد به این نتیجه می‌رسیم که مولد مبتنی بر FCSR از این نظر که حالت اولیه به کلید می‌باشد ایده‌آل نیستند.

٦–رمزگذارهای جویباری مبتنی بر FCSR ها

نظریه رمزگذاری جویباری با استفاده از FCSR ها بسیار تازه هستند. شاید کمتر بتوان نمونه‌هایی از این رمزگذارها را که پیاده‌سازی شده‌اند را پیدا کرد. در ادامه با مولدهای جویباری مبتنی بر FCSR که کاملاً شبیه به مولدهای ساخته شده با کمک LFSR، که در بخشهای قبلی بحث کردیم آشنا می‌شویم که در ساخت آنها هم از FCSR ها و LFSR ها استفاده می‌شود.

٦-١)مولدهای آبشاری

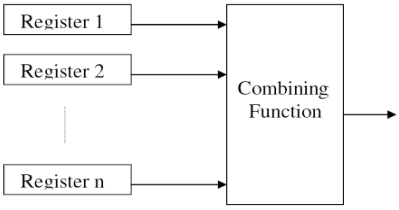
دو روش کلی در استفاده از FCSR ها در مولدهای آبشاری وجود دارد.

۱)آبشاری که فقط از FCSR ها ساخته شده باشد. آبشار Gollmann که قبلاً بحث شده با این تفاوت که به جای LFSR ها از FCSR ها استفاده می‌شود.

۲)آبشاری که از LFSR و FCSR ساخته شده است. آبشار Gollmann که به تناوب یک درمیان از LFSR و FCSR استفاده می‌شود.

٦-٢) مولدهای ترکیبی FCSR

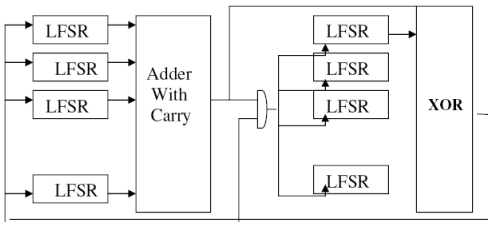
این مولدها با تعداد متغیری از FCSR ها یا ترکیبی از FCSR ها و LFSR ها ساخته می‌شوند. همچنین از یک سری توابع متنوع برای ترکیب آنها استفاده می‌شود. عملگر XOR خواص جبری FCSR ها را خراب می‌کند. پس در استفاده از عملگر XOR در ترکیبات FCSR ها حساسیتی وجود دارد. شکل (۱۳) این مولد را نشان می دهد. همانگونه که در شکل دیده می شود این مولد از تعداد متغیری از FCSR ها ساخته شده است. خروجی مولد نیز از XOR کردن خروجی‌های FCSR ها بوجود می‌آید.



شکل (۱۳) مولدهای ترکیبی

٦-٣) آبشار پریتی / مجموع FCSR / LFSR

تئوری جمع با نقلی ، ویژگیهای LFSR ها را از بین می‌برد و XOR نیز ویژگیهای جبری LFSR ها را از بین می‌برد. این مولد این ایده را ترکیب می‌کند. از این ایده‌ها در مولد مجمع FCSR / LFSR و در مولد پریتی FCSR / LFSR استفاده می‌شود. این مولد از یک سری از آرایه‌هایی از رجیسترها تشکیل شده است که در آن کلاک هر آرایه توسط خروجی آرایه قبلی کنترل می‌شود. شکل یک مرحله از مولد ترکیبی در زیر آمده است.



شکل (١٤)) آبشار پریتی / مجموع FCSR / LFSR

اولین آرایه از LFSR ها کلاک می‌خورند و نتایج آنها با استفاده از جمع کننده با نقلی با هم ترکیب می‌شوند اگر خروجی تابع ترکیب "1" باشد آرایه بعدی از FCSR ها کلاک می‌خورند و خروجی این FCSR ها با خروجی تابع ترکیبی قبلی با هم XOR می‌شوند. اما اگر خروجی تابع ترکیبی برابر "0" باشد در آنصورت آرایه FCSR ها کلاک نمی‌خورند. و خروجی فقط با نقلی نوبت قبلی جمع زده می‌شود. اما اگر خروجی دومین تابع ترکیبی برابر "1" باشد آرایه سوم از LFSR ها کلاک می‌خورد و به همین ترتیب ادامه پیدا خواهد کرد. این مولد از تعداد زیادی رجیستر تشکیل شده است. که کل رجیسترها برابر n*m خواهد بود که n نشان دهنده تعداد مرحله‌ها و m برابر تعداد رجیستر در هر مرحله می‌باشد. ولی در بهترین حالت توصیه می‌شود که n=10 و m=5 درنظر گرفته شود.

۷- نتیجه گیری

در این مقاله ما اصول کلی رمزگذاری جویباری را بیان نمودیم و رمزگذاری جویباری را با رمرگذاری بلاکی تا حدودی مقایسه نمودیم. همانگونه که بیان شد این نوع رمزگذاری دارای سرعت و قدرت بسیار زیادی می باشد . با استفاده از LFSR ها و FCSR ها می توان مولدهای بسیار قدرتمندی را طراحی نمود. انتقال واحدهای کوچک پیغام با استفاده از یک رمزگذار جویباری یک جریان کلید تولید می‌کند و رمزنگاری با ترکیب جریان کلید و پیغام (معمولاً عملیات ترکیب XOR است) انجام می‌پذیرد. تولید جریان کلید می‌تواند وابسته به پیغام و متن رمزی باشد (رمزگذارهای جویباری همزمان) یا می‌تواند به داده و رمزنگاریش وابسته باشد (رمزگذارهای جویباری خود همزمان). بیشتر طراحیهای رمزگذارهای جویباری برای رمزگذارهای جویباری همزمان صورت می‌گیرد. بدلیل خواص تئوری جذاب One-time pad، علایق زیادی نسبت به رمزگذارهای جویباری مشاهده می‌شود، اما هنوز همانند رمزگذارهای بلاکی یک استانداردی برای رمزگذارهای جویباری ارائه نشده است.

مراجع

[1] Hasan Dage , umut utkan ,"An xml based Data Exchange Model for power systemstudies "

[2] Common Format for Exchange of solved load Flow Data , Technical Report 6 (1973)

[3] چارلز اشباکر مترجم علیرضا زارع پور

اموزش برنامه نویسی xml در ۲٤ ساعت

[4] A.devose , S.E.widergren , J.Zhu , "xml For CIM Model Exchange"

[5] Xiao feng wang , Noel N.schulz , scott Neumann , "CIM Extentions to Electrical Distribution and CIM XML for the IEEE Radial test Feeders"

[6] Report on the common Information Model (CIM) Extensible Markup language(xml) Interoperability test #4 the power of the CIM to exchange power Model , EPRI.

[7] Drew Baigent , Mark Adamiak , Ralph Mackiewicz , "IEC 61850 Communication Networks and system In Substations".

[8] Karlheinz Schwarz "Standard IEC 61850 for substation Automation and other power system applications"

[9] Carstem Bose , Thomas Weber , Michael Sshwan , "Protection system coordination- considering New Techniques By Flexible Models"

[10] Carlos Martinez , Henry Huang, Ross Guttromson ,