پنجمین کنفرانس مهندسی برق ایران

۱۹-۱۷ اردیبهشت ۱۳۷۶

مجموعه مقالات - ۶

مخابرات (ب)

دانشگاه صنعتی شریف

دانشکده مهندسی برق

ICEE - 97

پنجمین کنفرانس مهندسی برق ایران ۱۹-۱۷ اردیبهشت ۱۳۷۶
Fifth Iranian Conference on Electrical Engineering May 7-9, 1997

روشی برای ایمن سازی سیستمهای کامپیوتری

با تلفیق روشهای RSQ و PRSQ

محسن شریفی، محمود فتحی و عباس خسرو بیگی - دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران

چکیده

در این مقاله ضمن بررسی معایب روش "سری اتفاقی" (RSQ - Random Sequence) در Encryption، به ارائه روشی برای رفع این معایب می‌پردازیم که ایمنی بیشتری را در حفاظت اطلاعات اختصاصی (Private information) ایجاد نماید. در روش پیشنهادی ضمن توضیح نحوه تقسیم سیستم کدگذاری به چند بخش، الگوریتم‌هایی از قبیل ایجاد پیچیدگی در نحوه انتخاب کلیدهای کدگذار، و یا تولید کلیدهای شبه اتفاقی (Pseudorandom) مطرح می‌شوند. بررسی آماری نشان میدهد که این روش احتمال شکست کمتری نسبت به روش "سری شبه اتفاقی" (PRSQ- Pseudorandom Sequence) دارد.
در مقایسه با روش RSQ، این روش از تعداد محدودتری کلید اتفاقی استفاده می‌نماید ولی مشابه روش PRSQ، سری کلیدهای شبه اتفاقی را با استفاده از آنها ایجاد می‌نماید. در ضمن این روش ازالگوریتمهای متغیری در قسمت‌های مختلف سیستم استفاده می‌کند که ایمن بودن آن را افزایش می‌دهد.

کلمات کلیدی

ایمنی، حفاظت، سری اتفاقی، سری شبه اتفاقی، رمزکردن، کلید، رمزگشایی، کشف رمز
Security, Encryption, Decryption, Random Sequence (RSQ), Pseudorandom Sequence (PRSQ), Key Generator, DES

۱- مقدمه

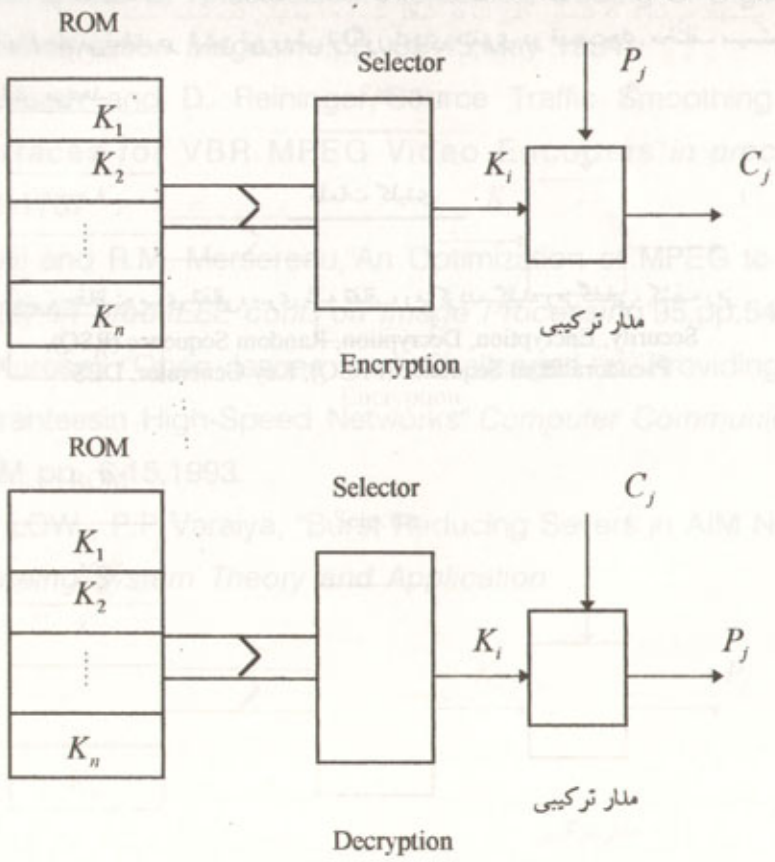
عصر کنونی قرن "تکنولوژی اطلاعات" نامیده شده است. تکنولوژی اطلاعات امروزه نقش اساسی در پیشرفت جوامع بشری دارد. در سیستمهای صنعتی، تجاری، علمی، تحقیقاتی و امنیتی، ایجاد امنیت در نگهداری و جلوگیری از دستیابی غیر مجاز به اطلاعات از اهمیت بالایی برخوردار است. از دیرباز تاکنون روشهای مختلفی برای برقراری امنیت ابداع و پیادهسازی شده است که هر یک دارای مزایا و معایبی بودهاند. یکی از این روشها، روش سری اتفاقی Random Sequence (RSQ) میباشد.

برای رفع اشکال این روش (حجم زیاد کلیدهای مورد نیاز برای Encryption)، روش سری شبه اتفاقی (Pseudo Random Sequence) PRSQ پیشنهاد شده است که بکارگیری آن موجب کاهش ضریب ایمنی سیستم میگردد. لازم بذکر است که این روشها میتوانند به دو صورت Bit Oriented و یا Character Oriented مورد استفاده قرار گیرند[1,3,4].

روش پیشنهادی در این مقاله Character Oriented میباشد و با استفاده از یک مدار پایه سختافزاری، که در کاربردهای عملی میتواند گیتهایی نیز به آن افزوده شود، پیادهسازی شده است. همانگونه که در این مقاله نشان داده میشود، این روش میتواند از روش PRSQ ایمنتر باشد.

۲- طرح اولیه

در یک سیستم ارسال و دریافت یا ذخیره و بازیابی اطلاعات، یکی از روشهای ایمنسازی، تعویض کدها با استفاده از یک روش کدگذاری موسوم به Encryption میباشد. برای این منظور میتوان از روش RSQ استفاده کرد. در این روش متن اصلی با یک سری کلیدهای اتفاقی ترکیب شده و متن کد شده بدست میآید. این سری کلیدهای اتفاقی برای ایجاد امنیت بالا بایستی یکبار مصرف باشند، در غیر اینصورت با داشتن یک نسخه متن اصلی و رمز شده آن کلیدهای مورد استفاده برای رمزگذاری براحتی بدست میآیند. از طرفی اگر کلیدها یکبار مصرف باشند موقع دریافت یا بازیابی، نیاز به حجم زیادی از کلید جدید خواهیم داشت. برای رفع اشکال فوق روش زیر پیشنهاد میشود که شمای کلی آن در شکل شماره ۱ نمایش داده شده است.



شکل شماره ۱: طرح اولیه روش پیشنهادی برای **Encryption**

در این طرح کلیدها در داخل یک حافظه ROM نگهداری می‌شوند. ROM می‌تواند از نوع Programmable باشد که در مواقع دلخواه کلیدها قابل تغییر باشند. مسلم است که تغییر کلیدها بایستی در فاصله‌های زمانی ثابت یا متغیری الزاما انجام شود. ضمنا برای حفاظت ROM می‌توان پس از اتمام کار با Encryptor یا Decryptor آنرا از روی مدار برداشت.

مدار Selector طرح فوق (شکل ۱) که وظیفه انتخاب یکی از n کلید موجود در حافظه را دارد، می‌تواند ترکیبی از انتخاب ۱ تا n را مورد استفاده قرار دهد. بعنوان ساده‌ترین راه می‌توان انتخاب مرتب از ۱ تا n را مورد استفاده قرارداد. یعنی از یک شمارنده ساده بالا رونده (Up Counter) برای آدرس‌دهی ROM استفاده نمود. تعداد بیتهای شمارنده باید حداقل برابر m باشند بطوریکه $n <= 2^m$ باشد. ضمنا بایستی شمارنده پس از عدد n مجددا عدد ۱ را بشمارد.

P_j کاراکتر j ام از متن اصلی می‌باشد. نرخ ورودی به مدار Encryptor بایستی با فرکانس شمارش شمارنده برابر باشد. بعنوان مثال اگر فرکانس ساعت شمارنده 1 HZ باشد بایستی در هر ثانیه یک کاراکتر از متن اصلی وارد مدار Encryptor شود.

مدار ترکیبی وظیفه ترکیب P_j ها را با K_i ها برای ایجاد C_j دارد، که به آن مانند یک جعبه سیاه نگاه می‌شود. در داخل این جعبه هرگونه عملیاتی می‌تواند بدلخواه انجام شود و بسته به کاربرد می‌تواند از پیچیدگی مناسبی برخوردار باشد.

C_j کدهای خروجی یا بعبارت دیگر کاراکترهای متن رمز شده می‌باشند.

در مدار Decryptor مجموعه کلیدهای ذخیره شده در ROM بایستی همانهایی باشند که در مدار Encryptor استفاده شده‌اند. همچنین مدار Selector رمزگشا دقیقا بایستی همان Function را داشته باشد که در موقع رمزگذاری داشته است. مسئله فرکانس انتخاب کلید Selector و نرخ ورودی C_j نیز مانند حالت رمزگذاری بایستی مدنظر قرار بگیرد. اگر تابع عملیاتی انجام شونده در جعبه مدار ترکیبی در موقع Encryption را F در نظر بگیریم، تابع مورد استفاده در مدار ترکیبی Decryptor بایستی F^{-1} باشد. یعنی عکس عملیات رمزگذاری را برای رمزگشایی انجام دهد. بعنوان ساده‌ترین مثال می‌توان از مدار XOR در هر دو طرف استفاده کرد.

$$P_j \oplus K_i = C_j \qquad \text{Encryption}$$

$$C_j \oplus K_i = P_j \qquad \text{Decryption}$$

۳- روشهای افزایش ایمنی در طرح اولیه

برای افزایش امنیت طرح فوق می‌توان با اضافه کردن یک مدول دیگر به آن، بصورت زیرعمل کرد:

۱- کلیدی را از میان سری کلیدهای اتفاقی (اولیه) انتخاب می‌کنیم.

۲- تعدادی دلخواه سری کلیدهای شبه اتفاقی (Pseudorandom) را با استفاده از کلید انتخابی در مرحله اول می‌سازیم.

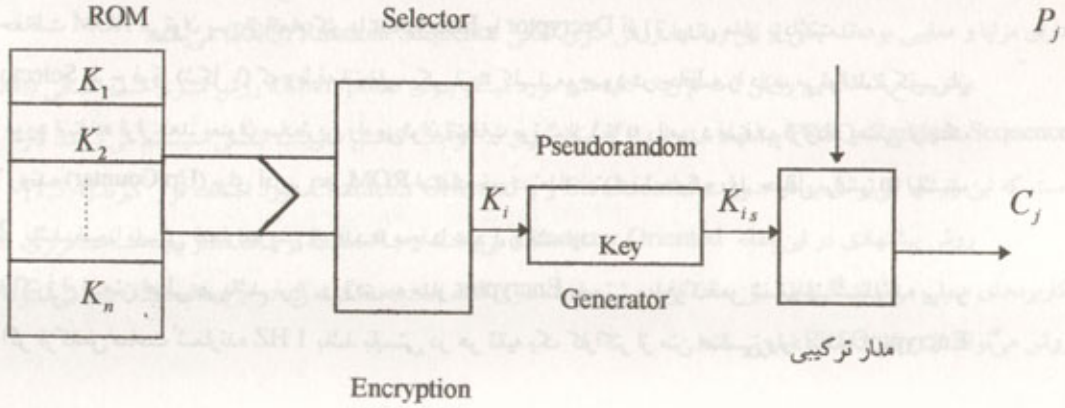
۳- تک تک کلیدهای بدست آمده در مرحله دوم را به ترتیب برای کدکردن کاراکترهای متن اصلی استفاده می‌کنیم.

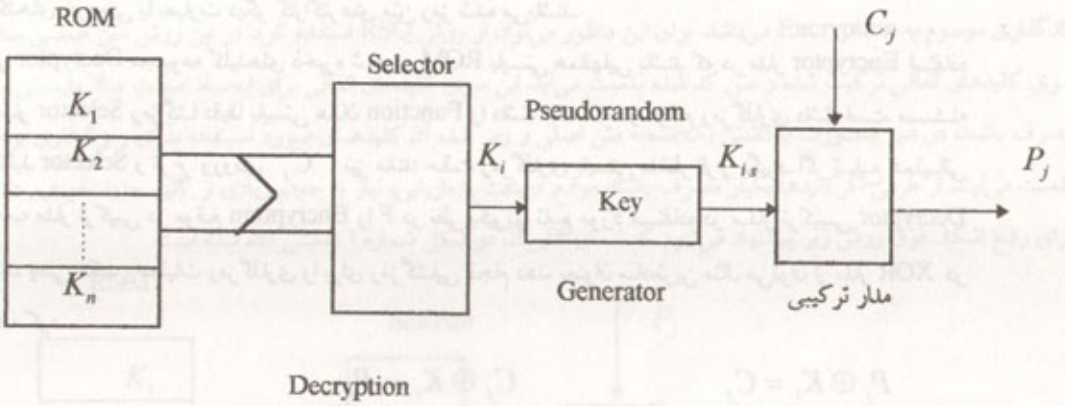
۴- به مرحله اول باز می‌گردیم و این عمل را تا پایان متن اصلی دنبال می‌کنیم.

لازم به ذکر است که مدار سخت‌افزاری مورد استفاده برای انجام عمل Encryption بایستی مجهز به خط کنترل EncrEn برای فعال یا غیر فعال ساختن مدار کدگذار باشد. در شکل شماره ۲ مدار این طرح نمایش داده شده است.

همه موارد ذکر شده در رابطه با طرح اولیه در این طرح نیز صادق است. علاوه بر این، در این طرح تعداد کاراکترهای P_j ورودی به مدار در واحد زمان بایستی برابر تعداد کلیدهای K_i ورودی به مدار ترکیبی در واحد زمان باشند. تعداد کلیدهای خروجی مدار Selector یعنی K_i بستگی به تعداد کلیدهای Pseudorandom تولید شده در جعبه Pseudorandom Key Generator دارد. مثلا اگر ما به ازای هر کلید اتفاقی ۱۶ کلید شبه اتفاقی تولید کنیم، در اینصورت تعداد کلیدهای K_i لازم برابر ۱/۱۶ کلیدهای K_{is} مورد نیاز می‌باشد. اینجا این نکته را نیز اضافه کنیم که اگربخواهیم روش را به صورت Bit Oriented

پیاده‌سازی کنیم، می‌توانیم در Pseudorandom Key Generator از بلوکهای (Feedback Shift Register) FSR، LFSR(Linear Feedback S.R.) و امثالهم استفاده کنیم [2].





شکل شماره ۲: طرح توسعه یافته پیشنهادی

همانطور که ملاحظه می‌شود، با اضافه کردن یک قسمت ساده به طرح اولیه، پیچیدگی عمل کدگذاری و کدگشایی چندین برابر شده و احتمال شکسته شدن الگوریتم را کاهش داده است.

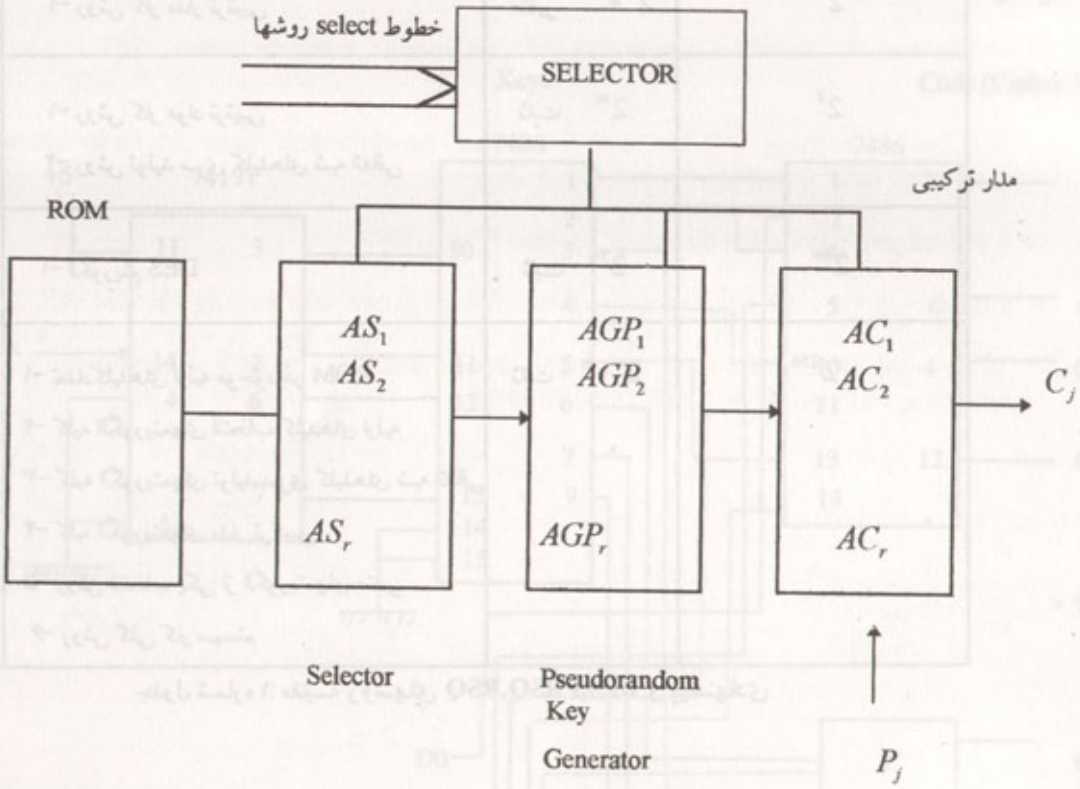
با توجه به اینکه یکی از راههای ممکن برای افزایش امنیت، ایجاد اختلاف ظاهری بین روش رمزگذاری و رمزگشایی در مبدا و مقصد یا در موقع ذخیره و بازیابی می‌باشد- مسلم است که در صورت وجود اختلاف در روش Encryption و Decryption با کشف یکی از این روشها نمی‌توان به ماهیت روش عکس آن پی برد- می‌توان این روش را نیز در طرحهای فوق به شرح ذیل اعمال کرد.

در طرحهای فوق لزومی ندارد که کلیدهای قرارگرفته در حافظه‌های مورد استفاده در Encryptor و Decryptor و روش Select آنها دقیقا مشابه هم باشند، بلکه می‌توان ترکیب‌های مختلفی را مورد استفاده قرار داد. (برای ذخیره کلیدها در حافظه و انتخاب یک کلید از مجموع n کلید) به گونه‌ای که خروجی K_i برای هر دو یکسان باشد. بعنوان مثال اگر در مبدا کلیدها به ترتیب از اول تا آخر Select می‌شوند، در مقصد کلیدها را از آخر به اول بچینیم و روش انتخاب نیز از آخر به اول باشد. همینطور می‌توان کلیدها K_i روش انتخاب و روش ایجاد کلیدهای K_{is} را طوری ترکیب کرد که در مبدا و مقصد در هر سه قسمت اختلاف موجود باشد، ولی نهایتا K_{is} های ایجاد شده در هر دو طرف یکسان باشند.

البته این مسئله در افزایش امنیت کلی سیستم چندان تاثیری ندارد، ولی با استفاده از این ترفند می‌توان در مواقع زمانی مشخصی در الگوریتم تغییراتی داد (که لزوما ایجاد این تغییرات در مبدا و مقصد همزمان نیست) و بدینوسیله بر پیچیدگی سیستم افزود.

راه دیگر افزایش امنیت به طرح اولیه، متغیر نمودن الگوریتم‌های انتخاب کلید از میان n کلید اولیه و یا تولید کلیدهای K_{is} از K_i ورودی به قسمت Pseudorandom Key Generator می‌باشد. بعنوان مثال می‌توان روش انتخاب یک کلید از میان n کلید موجود در حافظه را برای n کلید اول به ترتیب (از اول تا آخر) و برای n کلید دوم یک در میان و برای کلید سوم دو در میان و همینطور برای مابقی در نظر گرفت. همچنین می‌توان این کار را طوری انجام داد که در n کلید انتخاب شده، یک کلید از بین کلیدهای موجود در ROM چندبار استفاده نشود. با توجه به اینکه عملیات رمزنگاری و رمزگشایی در طرحهای فوق در چند سطح انجام می‌شود استفاده از چنین ترفندهایی باعث چندین برابر شدن پیچیدگی کلی سیستم خواهد شد.

در جای دیگری از طرح فوق نیز که می‌توان پیچیدگی سیستم را افزایش داد، مدار ترکیبی می‌باشد که P_j و K_{is} را گرفته و C_j ها را تولید می‌کند. این مدار ترکیبی نیز می‌تواند به صورت متغیر عمل کند. تعداد الگوریتم‌های متغیر در هر قسمت سیستم می‌تواند نامحدود باشد، ولی اگر محدود باشد می‌توان از یک Selector دیگر برای انتخاب روش مورد استفاده در هر قسمت جهت انجام عملیات محوله بهره جست. این روش در شکل شماره ۳ نمایش داده شده است.



شکل شماره۳: طرح نهایی

۴- تحلیل آماری طرح پیشنهادی

اگر فرض کنیم که طرف مقابل کلیه اطلاعات مربوط به سیستم رمز، غیر از خود کلیدها را داشته باشد، در روش PRSQ، اگر تعداد بیتهای کلید اولیه m تا باشد، با توجه به اینکه یک کلید m بیتی، 2^m حالت می‌تواند داشته باشد، برای دستیابی به متن اصلی نیاز به 2^m آزمایش می‌باشد [3] .

در روش RSQ، با فرض فوق، اگر طول پیام رمز شونده یا بعبارت دیگر تعداد کلیدها برابر I و تعداد بیتهای هر کلید برابر m باشد، حداکثر تعداد آزمایشات لازم برای دستیابی به متن اصلی 2^{m*I} می‌باشد.

در روش پیشنهادی اولیه مقاله با فرض فوق، اگر تعداد بیتهای هر کلید، m و تعداد کلیدها n باشد، تعداد آزمایشات لازم برابر 2^{n*m} خواهد بود. مقایسه این نتیجه با نتیجه مربوط به روش PRSQ خود گویای کارآیی این روش می‌باشد زیرا تعداد آزمایشات لازم برای کشف رمز در این روش $2^{m*(n-1)}$ برابر تعداد آزمایشات لازم در روش PRSQ می‌باشد.

بعنوان مثال اگر $n = 10$ و $m = 8$ باشد، تعداد آزمایشات لازم 2^{72} برابر تعداد آزمایشات لازم برای روش PRSQ خواهد بود. تعداد کل آزمایشات لازم در این حالت 2^{80} خواهد شد.

در نتیجه روش پیشنهادی کارآتر از روش PRSQ یا DES (Data Encryption Standard) می‌باشد. مقایسه این روشها در جدول شماره ۱ ذکر شده است.

روش	اطلاعاتی که بایستی تحلیل‌گر داشته باشد	حداکثر تعداد آزمایشات لازم	مثال: I= 1024 m= 8 n= 32
RSQ	۱- روش کار مدار ترتیبی	متغیر 2^{I*M}	2^{8192}
PRSQ	۱- روش کار مواد ترتیبی ۲- روش تولید سری کلیدهای شبه اتفاقی	ثابت 2^{m}	2^{8}
DES	۱- الگوریتم DES	ثابت 2^{56}	2^{56}
پیشنهادی	۱- تعدادکلیدهای اولیه موجود در ROM ۲- کلیه الگوریتمهای انتخاب کلیدهای اولیه ۳- کلیه الگوریتمهای تولید سری کلیدهای شبه اتفاقی ۴- کلیه الگوریتمهای مدار ترکیبی ۵- روش انتخاب یکی از الگوریتمهای متغیر ۶- روش کلی کار سیستم	ثابت 2^{n*m}	2^{256}

جدول شماره ۱: مقایسه روشهای DES,PRSQ,RSQ و پیشنهادی

۱-۴- منحنی مقایسات

در این قسمت منحنی‌های مقایسه دو روش DES و پیشنهادی (تلفیق RSQ و PRSQ) آمده است. تعداد آزمایشات لازم برای شکستن روش رمزنگاری DES برابر 2^{56} [4,3] و تعداد آزمایشات لازم برای شکستن روش پیشنهادی با توجه به اینکه m=8 فرض شده است، برابر 2^{8n} می‌باشد.

منحنی ۱ تعداد آزمایشات لازم برای شکستن روشهای مذکور به ازای $1 \le n \le 8$ را نشان می‌دهد. منحنی ۲ به ازای $1 \le n \le 16$ رسم شده است. منحنی ۳ زمان مورد نیاز برای شکستن الگوریتم رمز بر حسب n را نشان می‌دهد. این زمانها با فرض اینکه برای انجام هر آزمایش و بررسی نتیجه، تنها یک ثانیه وقت تلف شود، محاسبه شده‌اند.

منحنی‌های ۴ و ۵ و ۶ همان منحنی‌های ۱ و ۲ و ۳ می‌باشند که در آنها محور y به صورت لگاریتمی تقسیم‌بندی شده است.

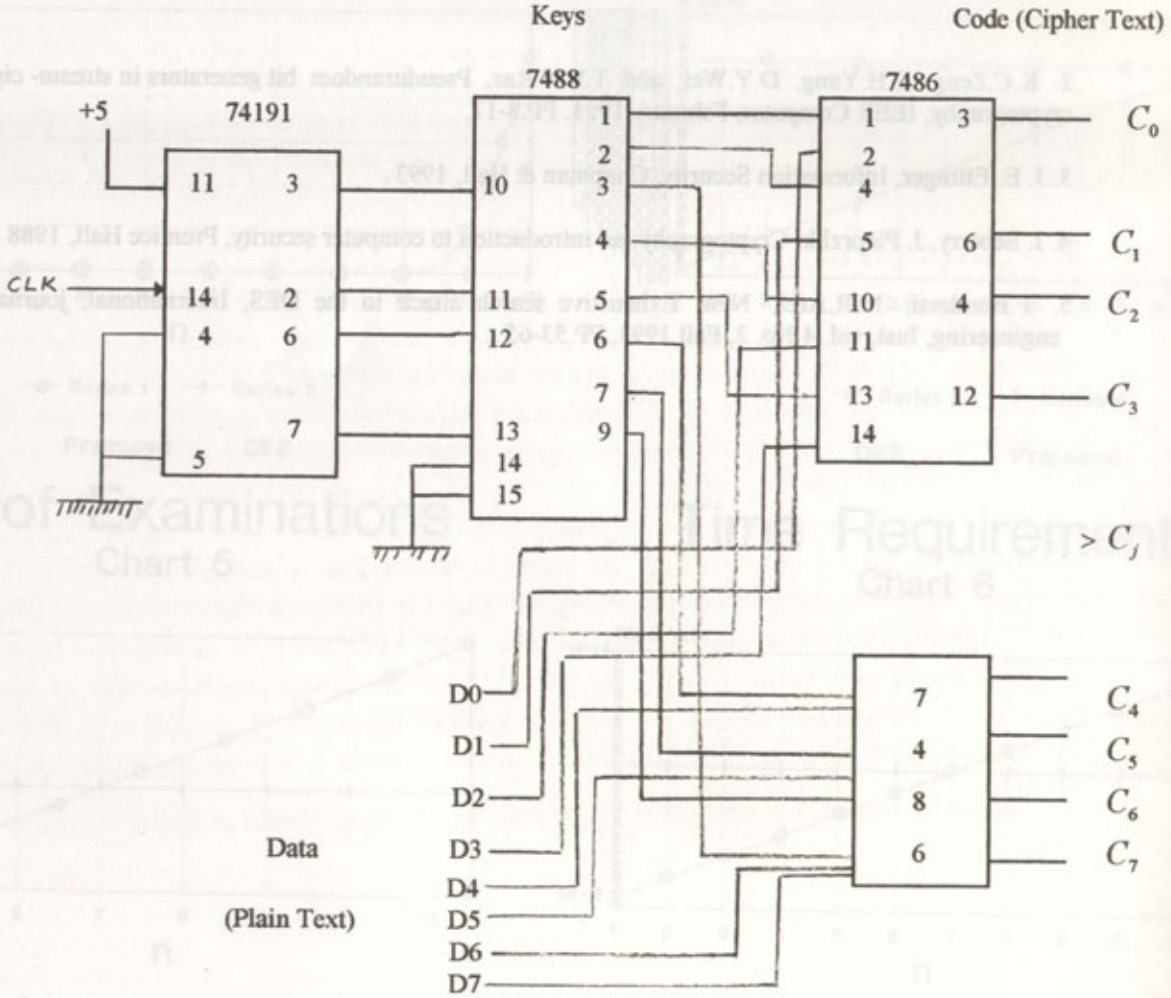
۵- پیاده‌سازی

پیاده‌سازی هر یک ازروشهای رمزنگاری می‌تواند به صورت سخت‌افزاری یا نرم افزاری باشد. مزایا و معایب هر یک از روشهای سخت‌افزاری و نرم‌افزاری به هیچکس پوشیده نیست. عمده‌ترین مزیت سخت‌افزار سرعت، وعیب آن، قیمت می‌باشد در عوض نرم‌افزار کند، ارزان و انعطاف‌پذیر است. انتخاب روش پیاده‌سازی بستگی به هزینه و زمان اجرا دارد.

۵-۱- پیاده‌سازی سخت‌افزاری

برای پیاده‌سازی سخت‌افزاری می‌توان از ICهای موجود استفاده کرد و یا پیاده‌سازی به صورت VLSI باشد. پیاده‌سازی سخت‌افزاری این روش محدودیت خاصی ندارد. با توجه به عدم امکان پیاده‌سازی VLSI در ایران، می‌توان با استفاده از ICهای موجود طرح فوق را پیاده‌سازی کرد. مسلم است که مدار ساخته شده با استفاده از VLSI سریعتر از مداری که از چندین IC استفاده کند، خواهد بود.

قسمتی از طرح عملی در شکل شماره ۴ نشان داده شده است، که بدلیل ساده بودن و بنیادی بودن مدار عملی ساخته شده، شکل کلی نشان داده شده است.



شکل شماره ۴: قسمتی از طرح عملی

هزینه ساخت این مدار با توجه به قیمت‌های موجود بازار قطعات الکترونیکی و کامپیوتری در ایران در حدود ۱۵۰۰۰ ریال خواهد شد که از قیمت بسیاری از سخت‌افزارهای آماده برای رمزنگاری کمتر می‌باشد.

۶- نتیجه

با بررسی روش Random Sequence جهت کاهش حجم کلیدهای مورد نیاز، روش پیشنهادی مطرح گردید. سپس با اضافه نمودن سطوحی به سیستم با استفاده از الگوریتم‌های متغیری در هر سطح، اصلاحاتی صورت گرفت. با توجه به چند سطحی شدن مدار و با استفاده از الگوریتم‌های متغیر در هر سطح، درجه ایمنی کل سیستم افزایش فوق‌العاده یافت. تحلیل آماری نشان می‌دهد که این روش کاملا موثر بوده و با سخت‌افزار ساده‌ای نیز پیاده‌سازی قابل انجام می‌باشد.

مراجع

1. D. Elizabeth, R. Denning, Cryptography and Data Security, Addison- Wesley, 1983.

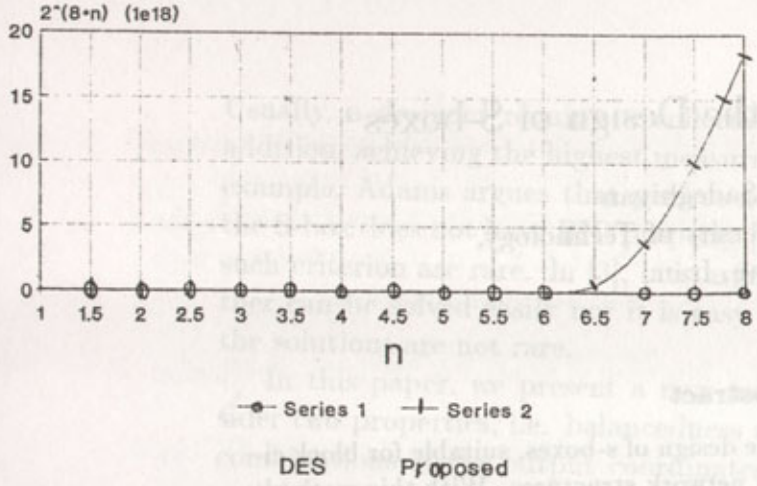
2. K.C.Zeng, C.H.Yang, D.Y.Wei, and T.R.N.Raa, Pseudurandom bit generators in stream- cipher cryptography, IEEE Computer, February 1991, PP.8-17.

3. J. E. Ettinger, Information Security, Chapman & Hall, 1993.

4. J. Seberry, J. Pieprzhk, Cryptography: an introduction to computer security, Prentice Hall, 1988.

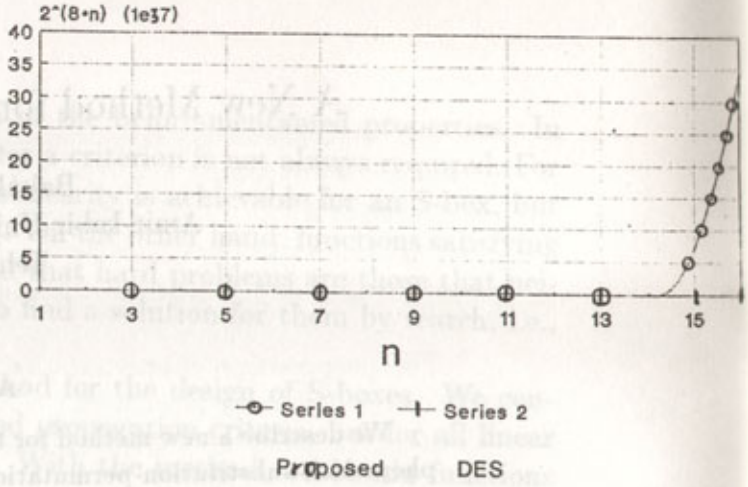
5. F.Hendessi, M.R.Aref, New Exhaustive search attack to the DES, International, journal of engineering, Iust, vol. 4 No. 2, Fall 1993, PP.53-65.

No. of Examinations
Chart 1



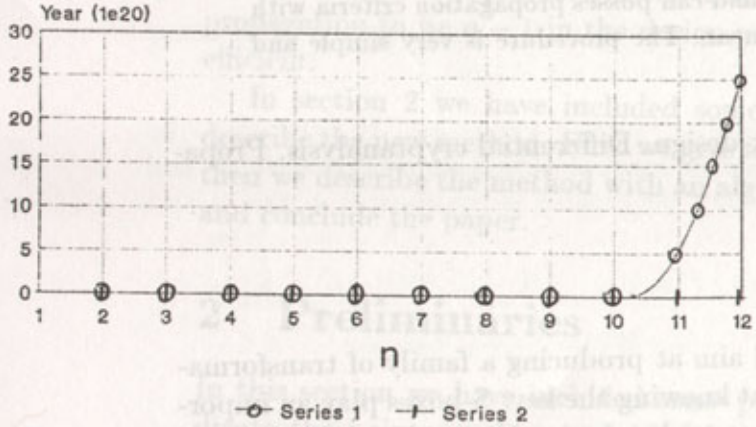
DES Proposed

No. of Examinations
Chart 2



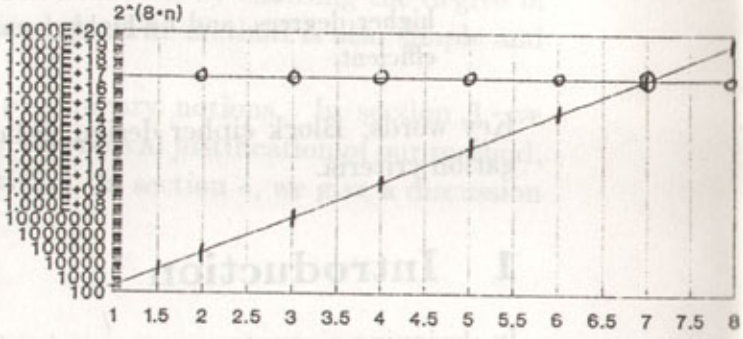
Proposed DES

Time Requirement
Chart 3



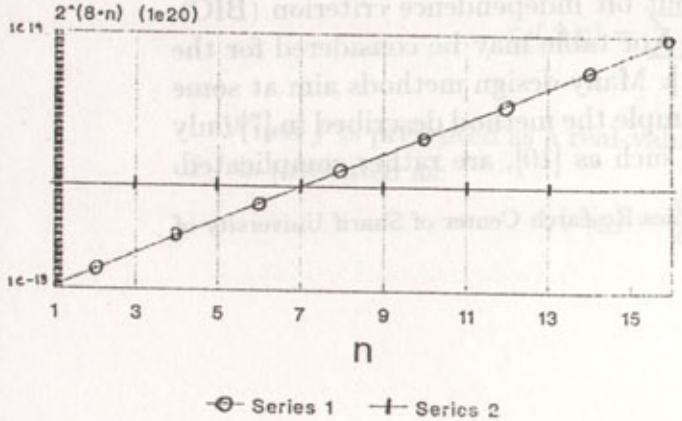
Proposed DES

No. of Examinations
Chart 4



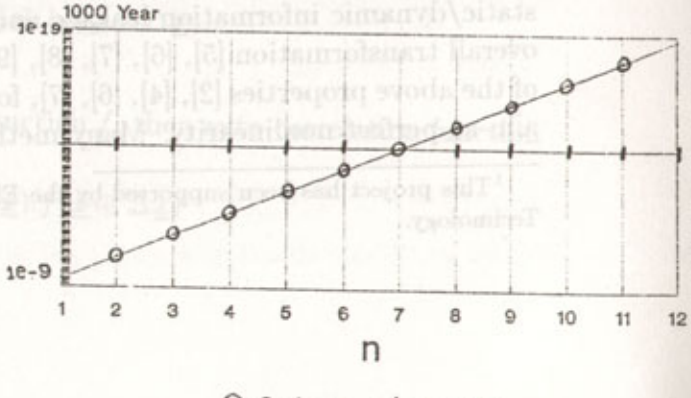
DES Proposed

No. of Examinations
Chart 5



Proposed DES

Time Requirement
Chart 6



Proposed DES