پنجمین کنفرانس انجمن رمز ایران
انجمن رمز ایران
Iranian Society of Cryptology
۲۴ الی ۲۶ مهر ماه ۱۳۸۶ دانشگاه علم و صنعت ایران

تحلیل نسخه خلاصه شده AES بوسیله الگوریتم ژنتیک

حسن توکلی
دانشگاه صنعتی خواجه نصیر الدین طوسی
hasantavakoli1983@gmail.com

علیرضافاتحی
دانشگاه صنعتی خواجه نصیر الدین طوسی
fatehi@kntu.ac.ir

محمود احمدیان
دانشگاه صنعتی خواجه نصیر الدین طوسی
m_ahmadian@kntu.ac.ir

بهروزحاجیان
دانشگاه صنعتی خواجه نصیر الدین طوسی
behrouz.hajian@gmail.com

چکیده: در این مقاله یک روش جدید جهت یافتن کلید در حالت جستجوی کامل فضای کلید ارایه شده است. این روش مبتنی بر تکنیک های جستجوی هوشمند می باشد. روشهای هوشمند در بهینه سازی این جستجو کاربرد های فراوانی دارند. در رمزنگاری با توجه به تعریف حملات مختلف می توان با دو دید به استفاده از سیستم های هوشمند به حملاتی با کارایی بالاتر رسید. اول آنکه خود حمله مبتنی بر سیستم های هوشمند باشد و یا ثانیا بهینه سازی حملات مختلف که در این مقاله از گزینه اول استفاده شده است.

واژه های کلیدی: رمزنگاریAES ، الگوریتم ژنتیک ، تحلیل هوشمند ،تحلیل خطی ،تحلیل تفاضلی

۱- مقدمه

در این مقاله یک روش جدید جهت یافتن کلید در حالت جستجوی کامل فضای کلید ارایه شده است. این روش مبتنی بر تکنیک های جستجوی هوشمند می باشد.اصولا در یک مساله بهینه سازی فارغ از ساختار مساله می توان همواره با جستجوی کامل فضا بمنظور یافتن اکسترمم ها اطمینان حاصل نمود.

روشهای هوشمند در بهینه سازی این جستجو کاربرد های فراوانی دارند. در رمزنگاری با توجه به تعریف حملات مختلف می توان با دو دید به استفاده از سیستم های هوشمند به حملاتی با کارایی بالاتر رسید. اول آنکه خود حمله مبتنی بر سیستم های هوشمند باشد در این حالت عموما منظور بهینه سازی جستجوی یک مشخصه مثلا جستجوی کامل فضای کلید بوسیله سیستم های هوشمند است. دیدگاه دوم بهینه سازی حملات مختلف مبتنی بر مسایل ریاضیات بوده و مساله اساسی در آنها بیشتر مبتنی بر یافتن یک مشخصه خاص است.

به عنوان مثال در سیستم های بلوکی مهمترین عامل یافتن مشخصات تفاضلی و خطی خوب از نظر تحلیل ودر سیستم های پی در پی یافتن همبستگی بین بخش های مختلف است.

هدف نهایی این مقاله تحلیل سیستم رمز بلوکی خلاصه شده AES [1] که نسخه خلاصه شده و آزمایشکاهی از سیستم رمز

[1] Simplified AES

بلوکی AES می باشد که بارویکرد اول مورد بررسی قرار می گیرد.

با توجه به مطرح شدن حملات خطی[1] و تفاضلی [2] بر ساختارهای مشابه فایستلی و مشابه SPN از جمله DES[1] و پس از فراخوان عمومی NIST[2] برای جایگزین نمودن یک سیستم رمز بلوکی و انتخاب نهایی و استاندارد شدن RINDAEL [3] به عنوان AES[3] حملات مختلف و متفاوتی مبتنی بر ارتقای حملات خطی و تفاضلی به این سامانه جدید مطرح شد. هدف اصلی این حملات اصولا یافتن یک ضعف در زیر ساختارهای این سیستم ها می باشد. ولی سوال اصلی این است که اگر ساختاری دارای زیر ساخت های غیر قابل رخنه باشد آنگاه چه باید کرد؟ طراحی یک S_box خوب با جدول تفاضلات هموار و یا استفاده از عوامل غیر خطی در انتهای یک دور الگوریتم و ... که منجر به مقاومت زیاد AES در برابر حملات شده است بطوریکه عملا حملات خطی و تفاضلی منجر به جوابهای غیر قبول از لحاظ زمان و حافظه و اطلاعات مورد نیاز می گردد و یا پیچیدگی محاسباتی حمله در حد جستجوی ساده و کامل فضای کلید می شود.

نکته قابل تامل دیگر در یافتن یک مشخصه بهینه برای حملات خطی و تفاضلی است که عملا هیچگاه بهترین حالت برای یک الگوریتم یافت نشده است.و همواره یافتن یک مشخصه خوب دغدغه اساسی این حملات است.

یکی از رویکرد های اخیر استفاده از ساختارهای هوشمند در جهت یافتن این مشخصات است که نتایج بهتری در چند سال اخیر بدست داده است.[4,5]

رویکرد دیگر استفاده از الگوریتم های هوشمند درجهت بهینه سازی جستجوی کلید است که در این مقاله صورت گرفته است.

ادامه این مقاله به این صورت تنظیم شده است در بخش ۲ به معرفی الگوریتم خلاصه شده AES می پردازیم و سپس در بخش ۳ به معرفی الگوریتم ژنتیک پرداخته شده است و در بخش ۴ به معرفی روش تحلیل جدید مبتنی بر الگوریتم ژنتیک اختصاص یافته ودر پایان نتایج بدست آمده با نتایج تحلیل خطی مقایسه شده است.

Simplified AES-۲

AES یک رمز بلوکی است که در طراحی آن سه معیار زیر در نظر گرفته شده است.

_مقاومت در برابر تمامی حملاتی که تا آن زمان شناخته شده بود
_سرعت مناسب و کد فشرده برای پهنه وسیعی از platform ها
_سادگی طرح

در بیشتر رمز های بلوکی از ساختار فایستل بعنوان تبدیل دور استفاده شده است اما در AES تبدیل دور از سه تبدیل یکنواخت و برگشت پذیر تشکیل شده است که آنها را لایه مینامیم. در این حالت هر لایه وظیفه خاصی دارد:

_لایه ترکیب خطی انتشار را در خلال چند دورتضمین می کند
_لایه غیر خطی استفاده از S_box هایی با خواص غیر خطی مناسب را فراهم می کند.
_لایه جمع با کلید یک عمل xor ساده در حالات میانی با کلیدهای دور را اجرا می کند.

تبدیل های هر دور ترکیبی از ۴ تبدیل متفاوت است. شبه کد آن بصورت زیر است :

```
Round(State,Roundkey)
{
        Bytesub(State);
        Shift Row(State);
        Mixcolumn(State);
        Add_Roundkey(State,Roundkey);
}
```

در دور نهایی رمز این تبدیل دارای شکل متفاوت می باشد که بصورت زیر است:

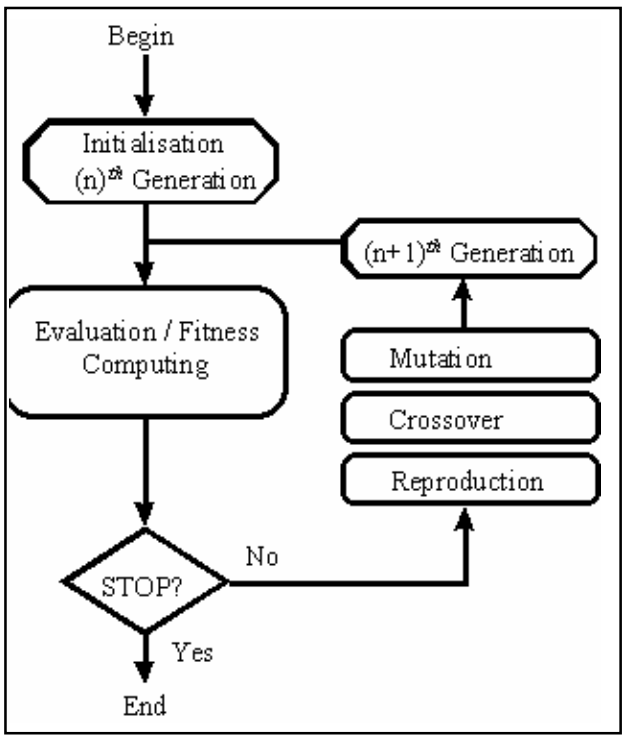
```
Final Round(State,Roundkey)
{
        Bytesub(State);
        Shift Row(State);
        Add_Roundkey(State,Roundkey);
}
```

Simplified AES اولین بار در [6] مطرح شد و در واقع نسخه خلاصه شده ای از AES واقعی می باشد و بمنظور آموزش نحوه یادگیری عملکرد AESواقعی مطرح شده است.

Simplified AES دارای لایه های یکسانی نسبت به AES واقعی می باشد با این تفاوت که طول کلید آن از ۱۲۸ بیت به ۱۶ بیت کاهش یافته است و به همان نسبت دارای لایه هایی با ابعاد کوچکتر است.

این الگوریتم در [7] مورد تحلیل خطی قرار گرفته است که مرجع مقایسه نتایج این مقاله می باشد.

[1] Data Encryption Standard
[2] National Institute of Standard Technology
[3] Adavanced Encryption Standard



شکل ۱ : نحوه انجام الگوریتم ژنتیک

در این حمله تعداد ۵۴۸ متن اصلی و معادل رمز شده آن بـرای تحلیل مورد احتیاج می باشد.

۳- الگوریتم ژنتیک

الگوریتم ژنتیک روش مناسبی برای کاربرد درحل مسائلی است که توام با جستجو و بهینه سازی می باشند. اصول کار این الگوریتم مبتنی برآنچه که در فرآیندهای بیولوژیکی رخ می دهد استوار می باشد. در طبیعت پس از گذشت سالها و انجام زاد و ولد جانداران در طی چند نسل , جمعیت های طبیعی بر مبنای اصول انتخاب طبیعی یعنی اصل بقا , سازگارترین ها توسعه می یابد.

روشهای غیر هوشمند جوابگوی درستی برای مسائلی مانند یافتن اکسترمم یک تابع ناهموار نمی تواند باشد چرا که این روش ها اولا نیاز به توصیف دقیقی از مساله داشته و ثانیا بعضی از مسائل خارج از قدرت بهینه سازی آنها میباشد. بنابراین واضح است که ابزار به کاررفته در رمز گشایی بایستی مبتنی بر مدلهای طبیعی بوده تا بتواند جای عملیات سری ویا ترتیبی موجود مسائل پیچیده وغیرخطی را به طریق هوشمند حل نماید.

همانطور که در شکل ۱ ملاحظه می کنید الگوریتم ژنتیک متشکل از ۳ اپراتور می باشد که در ادامه به معرفی آنها می پردازیم.

۳-۱-اصول اساسی الگوریتمهای ژنتیک

قبل از اعمال الگوریتم ژنتیک به یک مسئله لازم است که بیان یا کدینگ مناسبی برای مسئله پیدا کرد و علاوه بر این باید تابع برازندگی[1] مسئله مزبور را _ که میزان شایستگی هر حل کد شده را نشان می دهد _ نیز مشخص نمود .ضمن اجرای الگوریتم نیز باید والدینی برای انجام عمل "باز تولید" انتخاب نموده و عمل "آمیزش" را بطورمناسبی بین آنها انجام داد.

۳-۲-کدینگ

فرض می شود که راه حل یک مسئله را بتوان بصورت مجموعه ای از پارامترهای مطلوب نشان داد. از اتصال این پارامترها که بنام ژن می شناسیم زنجیره ای از مقادیر بوجود می آید که تحت عنوان کروموزوم شناخته می شود. در این مورد ما از کدینگی که مبتنی بر آمارگان(که ناشی از یک پیام اصلی نوعی و یک پیام رمزشده نوعی می باشد) بین بیت های کلید و تابع برازندگی می باشد و بطور آماری انتخاب گردیده است بهره می گیریم.

۳-۳-تابع برازندگی

بکارگیری الگوریتم ژنتیک در حل هر مسئله ای موکول به یافتن تابع برازندگی آن مسئله است. به ازای هر کروموزوم تابع برازندگی یک مقدار عددی می دهد که برازندگی آن کروموزوم نامیده می شود و توانایی یا سودمندی آن کروموزوم را مشخص می کند.

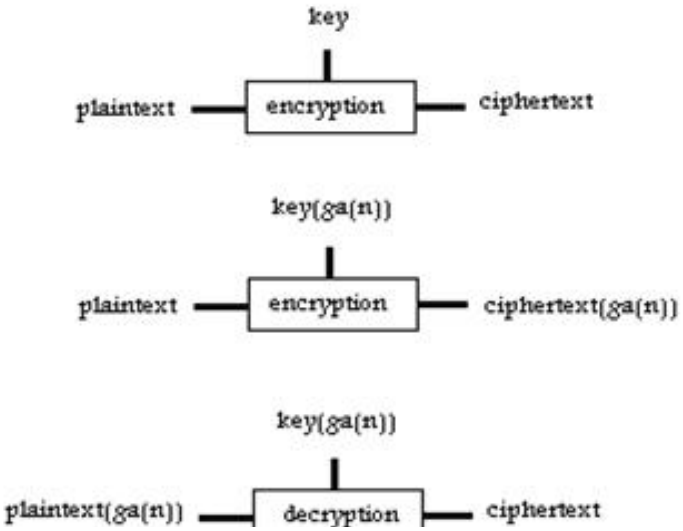
۳-۴- گزینش[2]

در مرحله باز تولید لازم است اعضایی از جمعیت قبلی انتخاب شوند تا برای تولید فرزندانی که نسل بعدی را تشکیل می دهند بین آنها آمیزش صورت گیرد. والدین بصورت تصادفی از بین اعضای جمعیت انتخاب می شوند بطوریکه آنهایی که برازندگی بیشتری دارند شانس بیشتری برای انتخاب شدن داشته باشند.

۳-۵-برش[3]

این اپراتور دو عضو را انتخاب کرده و زنجیره های کروموزوم های آنها را در یک محل انتخابی بصورت تصادفی قطع کرده تا هر کروموزوم به دو قسمت ابتدایی و انتهایی تقسیم شود.

[1] Fitness Function
[2] Reproduction
[3] Crossover



شکل۲ : (a) تولید plaintext,ciphertext مورد نیاز حمله
(b)تولید ciphertext بوسیله کلید تولیدی توسط الگوریتم ژنتیک
(c) تولید plaintext بوسیله کلید تولیدی توسط الگوریتم ژنتیک

سپس جای قسمت های انتهایی کروموزوم های تقسیم شده را با هم عوض می کند تا دو فرزند جدید تولید شود.

۳-۶-جهش[1]

این اپراتور پس از انجام تلاقی بطور جداگانه به هر کدام از فرزندان اعمال می شود و بطور تصادفی با یک احتمال خیلی کم هر ژن را متحول می کند. این اپراتور اصولا برای جلوگیری از بدام افتادن الگوریتم ژنتیک در یک نقطه اکسترمم محلی بکار می رود. برای تولید بردار آزمون از روش جهش باینری استفاده می نماییم که درآن ژنهای یک کروموزوم با یک احتمالی متمم می شوند.

۴-تحلیل Simplified AES بوسیله الگوریتم ژنتیک

_ طراحی حمله مبتنی بر جستجوی ساده در فضای کلید بمنظور یافتن کلید با فرض در اختیار داشتن یک متن اصلی ویک متن رمز شده معادل آن می باشد که اعداد از 0 تا $2^{|K|}-1$ به ترتیب یک به یک به عنوان کلید در الگوریتم قرار داده می شوند (|K| یعنی تعداد بیت های کلید) اگر جواب درست بود کلید یافت شده و گرنه کلید بعدی امتحان می شود. این حمله قطعا بلادرنگ نبوده و استخراج کلید با تاخیر همراه است. [8]

_ طراحی حمله مبتنی بر جستجوی هوشمند در فضای کلید بمنظور یافتن کلید که در این مقاله معرفی می گردد بدین گونه است که حمله بر اساس تنها یک متن اصلی و یک متن رمز معادل می باشد. بطور تصادفی تعدادی از کلید ها در فضای کلید انتخاب می شوند (POPULATION) که براساس یک کدینگ مناسب که در ادامه به معرفی آن خواهیم پرداخت دارای یک برازندگی هستند. سپس با انجام سه عملگر ژنتیکی برروی جمعیت قبلی و با توجه به تابع برازندگی همگرایی و یافتن جواب بهینه صورت می گیرد ودرنتیجه به سمت اکسترمم شدن تابع برازندگی حرکت می کند.

در این راستا با توجه به شکل ۲ بایدتوجه نمود که

_تابع برازندگی باید طوری انتخاب شود که به ازای کلیدی که مورد جستجو قرار می گیرد تابع برازندگی اکسترمم شود. مثال مناسب برای این تابع :

Fitness=F(x,y)
x=1-correlation(ciphertext(ga(n)), ciphertext)
y=1-correlation(plaintext (ga(n)) , plaintext)

[1] Mutation

که در آن F یک تابع اکیدا صعودی دلخواه از (x,y) می باشد و در عبارت بالا ga(n) یعنی پارامترهای ناشی از الگوریتم ژنتیک در مرحله n ام (generation در مرحله n ام) میباشد. باید درنظر گرفت که تابع برازندگی F باید یک تابع اکیدا صعودی باشد که مینیمم آن در F(0,0) قرار می گیرد.

در این حالت کدینگ باید طوری طراحی شود تا کلیدهای نزدیک به یکدیگر، توابع برازندگی نزدیک به یکدیگر را بوجود آورد. به بیان دیگر و با استفاده از ایده حمله تفاضلی در اینحالت می توان یک جدول تفاضلی بوجود آورد که در آن به جای تفاضلات پیام اصلی از تفاضلات بیت های کلید و به جای تفاضلات پیام رمز شده از تفاضلات تابع برازندگی استفاده نمود. این جدول به ازای یک متن اصلی و متن رمز شده مشخصی بوجود آمده است . فایده استفاده از چنین جدولی این است که می توانیم با هرکلیدی که توسط الگوریتم ژنتیک در مرحله n ام و با بدست آمدن تابع برازندگی آن کلید میتوانیم فاصله تابع برازندگی را یافته و کلید بعدی را تقریب بزنیم ودر نهایت به یک کدینگ در الگوریتم رسید که جستجو را همگرا می نماید.

در ادامه استفاده از الگوریتم ژنتیک ما از خاصیت وابستگی خطی (حمله خطی) بین بیت های پیام اصلی و بیت های پیام رمز شده وبیت های کلید که یکی از محدودیت های فضای جستجو بوده و فضای جستجو را به شدت کاهش می دهد استفاده نمودیم که همگرایی الگوریتم را به شدت تحت تاثیر قرار داد. این روابط همان روابط با احتمال بالا در [7] می باشند که ما به عنوان نمونه از ۱۰ رابطه استفاده نمودیم.

این حمله در حقیقت همان الگوریتم جستجوی کلید می باشد که بوسیله الگوریتم ژنتیک هوشمند شده است . این هوشمندی ناشی از استفاده ضمنی از حمله خطی جهت کاهش فضای جستجو که خود ناشی از رابطه خطی بین بیت های پیام اصلی و بیت های پیام رمز شده وبیت های کلید می باشد . همچنین از ایده حمله تفاضلی که منجر به ایجاد جدولی که در آن به جای تفاضلات پیام اصلی از تفاضلات بیت های کلید و به جای تفاضلات پیام رمز شده از تفاضلات تابع برازندگی استفاده می شود بهره برده ایم. این فرآیند تقریب مناسبی از بیت های بعدی کلید را میدهد و به نحوی باعث ایجاد یک کدینگ مناسب می گردد.

این روش که تلفیقی از استفاده تحلیل خطی و همینطور تحلیل تفاضلی می باشد به نتایج بهتری از هر یک از این دو تحلیل می انجامد.

در جدول ۱ به مقایسه بین مولفه ای تحلیل ارایه شده و همینطور تحلیل خطی [7] پرداخته ایم. همانطوریکه می بینیم حمله معرفی شده در این مقاله تنها احتیاج به یک متن اصلی و معادل رمز شده دارد و همینطور زمان تحلیل به نحو چشمگیری کاهش یافته در صورتیکه برای تحلیل خطی [7] ۵۴۸ متن اصلی و معادل رمز شده مورد احتیاج می باشد. همینطور در جدول ۲ به مقادیر مورد استفاده در این مقاله برای الگوریتم ژنتیک پرداخته شده است.

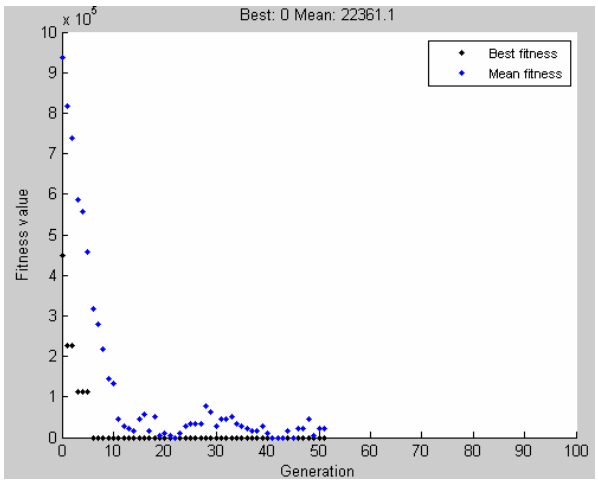
در شکل ۳ نیز به عنوان نمونه تغییرات تابع برازندگی الگوریتم ژنتیک آورده شده است.

جدول ۱ : مقایسه بین حمله ارایه شده و حمله [7]

	حمله معرفی شده	تحلیل خطی
متن مورد نیاز	1	548
حافظه موردنیاز	medium	medium/low
زمان مورد نیاز	Real time	With delay

جدول ۲ : پارامترهای الگوریتم ژنتیک

Population	**10**
Crossover	**0.6**
Mutation	**0.15**
Maximum Generation	**90**



شکل ۳: نمونه ای از تغییرات تابع برازندگی

۵- نتیجه گیری

در این مقاله یک حمله مبتنی بر ساختارهای هوشمند ارایه شد که در آن از هر دو ایده تحلیل خطی بمنظور کاهش فضای جستجو و همینطور از تحلیل تفاضلی برای ایجاد یک کدینگ و دسته بندی ورودی ها استفاده شد. نتایج شبیه سازی بیانگر کارایی این نوع تحلیل می باشد .در آینده می توان به گسترش حمله معرفی شده در این مفاله بر ساختارهای مشابه فایستلی با طول کلید ۱۲۸ و یا ۱۹۲ بیتی پرداخت.

۶- مراجع

[1] Matsui, M. "Linear cryptanalysis method for DES cipher". *Advances in Cryptology - EUROCRYPT 1993.*

[2] Eli Biham, Adi Shamir, "Differential Cryptanalysis of the Data Encryption Standard", Springer Verlag, 1993..

[3]J. Daemen and V. Rijmen, , "AES Algorithm" submission, September 3, 1999.

[4] Hernandez, J.C .; Isas i, P . " New results on the genetic cryptanalysis of TEA and reduced-round versions of XTEA". IEEE Proceeding Evolutionary Computation , 2004. CEC2004.Congress on Volume 2, Issue,19-23 June

[5]R.Spillman,M.Janssen,B.Nelson,M.Kepner "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers", Cryptologia, Volume XVII , Issue 1 (January 1993), Pages: 31 – 44,Year of Publication: 1993

[6] Schaefer, Musa and Wedig , "Simplified AES Algorithm and its Linear and Differential Cryptanalyses". *Cryptologia* 2003.

[7] Mansoori, S.D. Bizaki, H.K." Linear cryptanalysis on second round simplified AES", IEEE Proceeding 2006Volume: 2

[8] Andelman, D.; Reeds, J." On the cryptanalysis of rotor machines and substitution - permutation networks",Information Theory, IEEE Transactions on Volume 28, Issue 4, Jul 1982 Page(s): 578 - 584