مجله علمی پژوهشی «علوم و فناوری‌های پدافند نوین»
سال پنجم، شماره ۳، پاییز ۱۳۹۳؛ ص ۲۱۰-۲۰۳

شکستن الگوریتم رمزنگاری SDES با استفاده از الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده

میثم مرادی[۱]، حسن ختن لو[۲*]، مهدی عباسی[۳]

۱- گروه کامپیوتر، دانشکده فنی و مهندسی، واحد علوم و تحقیقات همدان، دانشگاه آزاد اسلامی واحد همدان

۲- دانشیار ۳- استادیارگروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی‌سینا

(دریافت: ۹۳/۰۴/۲۴، پذیرش: ۹۳/۰۶/۱۸)

چکیده

در سال‌های متمادی، شکستن الگوریتم‌های رمزنگاری به‌عنوان یک چالش مورد توجه قرار گرفته است. الگوریتم رمزنگاری DES به‌عنوان - استانداردی جهت محرمانه نگه‌داشتن اطلاعات مورد استفاده قرار گرفته و این امکان برای محققان وجود دارد که آن را بیازمایند. الگوریتم رمزنگاری SDES نسخه ساده شده الگوریتم رمزنگاری DES می‌باشد که محققان جهت پژوهش، الگوریتم رمزنگاری SDES را مورد استفاده قرار می‌دهند. در این تحقیق از الگوریتم استاندارد بهینه‌سازی پرتو ذرات، جهت شکستن الگوریتم رمزنگاری SDES استفاده شده است. جهت شکستن الگوریتم رمزنگاری SDES، الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده است. چندین حمله روی بلوک‌های از داده‌های متنی صورت گرفت که نتایج بدست آمده نشان می‌دهد این الگوریتم درکشف بیت‌های کلیداصلی درکوتاه‌ترین زمان بهبودعملکرد داشته به طوری که در مقایسه با کارهای پیشین در معیار کشف بیت‌های کلیداصلی، ده بیت کلیداصلی کشف شده است و در معیار زمان کشف بیت‌های کلیداصلی، زمان از بیش یک دقیقه به کمتر از بیست ثانیه کاهش یافته است.

کلید واژه‌ها: شکستن‌رمز، استاندارد رمزنگاری داده ساده شده، الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، کلید اصلی.

Breaking of Simplified-Data Encryption Standard Using Optimized SPSO

M. Moradi, H. Khotanlou*, M. Abbasi
Department of Computer Engineering, Engineering Faculty, Bu-Ali Sina University
(Received: 15/07/2013; Accepted: 09/08/2014)

Abstract

In many years Breaking of cryptography Algorithms have been taken into consideration as a challenge. Data Encryption Standard (DES) is used as a standard to keep the information confidential and allows the research studies to investigate it.S-DES is the simplified version of Data Encryption Standard Algorithm which is used by the researchers for investigation. The present study uses SPSO algorithm to break SDES algorithm. To break SDES algorithm the SPSO was optimized. There were some attacks on some blocks of the cipher text and the obtained results revealed that this algorithm had a better performance in detecting main key bits in shortest time so that, in detection factor main key bits, ten main key bits have been discovered and in the time of detection factor main key bits, time over a minute to less than twenty seconds dropped in comparison with Previous work.

Keywords: Decryption, Simplified-Data Encryption Standard (S-DES), SPSO, Main Key.

*Corresponding Author E-mail: Khotanlou@basu.ac.ir

۱. مقدمه

با توجه به افزایش حجم مستندات متنی و تبادل آن بین افراد در سطح‌های مختلف ارتباطاتی نیازمند الگوریتم‌های قدرتمند در زمینه رمزنگاری هستیم. با توجه به اینکه علم رمزنگاری و علم رمزشکنی مکمل یکدیگرند، علم رمزشکنی به محققان این امکان را می‌دهد که یک الگوریتم رمزنگاری بررسی شود و نقاط ضعف و قوت آن نمایان شود و به صورت تکاملی نقاط ضعف الگوریتم‌های رمزنگاری در مقابل الگوریتم‌های رمزشکنی ترمیم شود که این فرایند تکامل باعث به وجود آمدن الگوریتم رمزنگاری مطمئن و کارا و درنتیجه اطمینان خاطر افراد در تبادلات ارتباطی است.

سامانه رمزنگاری به دو رده کلی رمزنگاری کلید متقارن[1] و رمزنگاری کلید نامتقارن[2] تقسیم‌بندی می‌شود که در رمزنگاری کلید نامتقارن، رمزگشایی و رمزگذاری با دو کلید متفاوت انجام می‌شود در حالی که در رمزنگاری کلید متقارن، رمزگشایی و رمزگذاری باکلیدی مشابه انجام می‌شود. الگوریتم رمزنگاری DES[3] یکی از معروف‌ترین الگوریتم‌های سامانه رمزنگاری کلید متقارن محسوب می شود [۱]. الگوریتم SDES[4]، ضمن حفظ ساختار معماری الگوریتم رمزنگاری DES، به عنوان نسخه ساده شده این الگوریتم می‌باشد که در پژوهش‌های مختلف مورد استفاده قرار می‌گیرد. در این تحقیق، شکستن الگوریتم استاندارد رمزنگاری داده ساده شده بررسی شده است (SDES). در الگوریتم جستجوی حمله‌قوی[5] با آزمودن تمامی کلیدهای ممکن سعی شده است کلید واقعی که متن رمز شده را به متن اصلی تبدیل می کند را پیدا کند ولی برای حدس زدن کلید واقعی زمان زیادی صرف می‌شود. الگوریتم‌های فرااکتشافی [۲] دارای ساختار جستجوی تصادفی هستند که جهت شکستن الگوریتم رمزنگاری SDES استفاده شده است. درکارهای پیشین، زمان کشف بیت‌های کلید اصلی بیش از یک دقیقه بوده و کشف کامل بیت‌های کلید رمز در هیچ قالب متنی امکان‌پذیر نبوده است (عدم رمزگشایی). الگوریتم استاندارد بهینه‌سازی پرتو ذرات[6] به عنوان یکی از الگوریتم‌های فرااکتشافی، دارای ایده تکاملی است. در این تحقیق، با تنظیم دقیق پارامترهای الگوریتم استاندارد بهینه‌سازی پرتو ذرات و با طراحی الگوریتمی جهت شکستن الگوریتم رمزنگاری SDES، این الگوریتم بهینه شده است. با بهینه‌شدن الگوریتم استاندارد بهینه‌سازی پرتو ذرات درمقایسه با کارهای پیشین زمان کشف بیت‌های کلید اصلی کاهش یافته و در تمام بلوک‌های متنی قادر به کشف کامل بیت‌های کلید اصلی نموده و می‌توان متن‌اصلی را رمزگشایی کرد.

باقیمانده این تحقیق به شرح ذیل سازماندهی شده است: در بخش دوم به بررسی کارهای مرتبط و مقایسه روش پیشنهاد شده با کارهای پیشین پرداخته شده است. در بخش‌سوم، مروری بر الگوریتم رمزنگاری SDES، در بخش‌چهارم، توصیف الگوریتم استاندارد بهینه‌سازی پرتو ذرات به‌کار گرفته شده در این تحقیق، جزئیات مراحل مختلف بهینه کردن الگوریتم استاندارد بهینه‌سازی پرتو ذرات پرداخته شده است. در بخش‌پنجم، نتیجه و بحث به منظور ارزیابی روش پیشنهاد شده در مقایسه با کارهای پیشین و در بخش‌ششم نتیجه‌گیری مقاله با اشاره به پیشنهاداتی برای گسترش در آینده بررسی شده است.

۲. کارهای مرتبط

در سال‌های اخیر پژوهش‌های متعددی در حوزه تحلیل رمز انجام شده است. عبد المونیم و همکاران[۳] حمله به DES16 را با استفاده از الگوریتم PSO انجام دادند. آکیویت و همکاران [۴] شبکه عصبی مصنوعی را جهت تحلیل رمز الگوریتم رمزنگاری DES به کار گرفتند. الالایح و همکاران [۵] حمله به الگوریتم رمزنگاری SEDS را با استفاده از شبکه عصبی تحلیل کردند. سومان و همکاران [۶] توسعه سیستم رمزی را با استفاده از الگوریتم SDES از طریق جانشینی متون رمزی به کار گرفتند. الشکارچی [۷] تحلیل‌رمز الگوریتم DES را با استفاده از شبیه‌سازی شبکه عصبی مصنوعی انجام دادند. گارج [۸] الگوریتم ترکیبی[7] و الگوریتم ژنتیک را جهت تحلیل رمز SDES مقایسه کرد. الانازی و همکاران [۹] مطالعه‌ای روی الگوریتم‌های رمزنگاری 3DES، DES و AES داشتند. حسن حسین و همکاران [۱۰] الگوریتم ژنتیک را جهت تحلیل رمز DES8 استفاده کردند. شارما و همکاران [۱۱] مطالعه‌ای روی عملکرد الگوریتم DES و SDES داشتند. جادن و همکاران [۱۲] تحلیل‌رمز DES را با استفاده از الگوریتم بهینه‌سازی پرتو ذرات باینری انجام دادند. کنژ و همکاران [۱۳] یک گزارش روی تحلیل رمزهای مختلف ارائه دادند. صلابت و همکاران [۱۴] الگوریتم کلونی مورچه‌ها را جهت تحلیل‌رمز DES استفاده کردند. ستهیا و همکاران [۱۵] یک روش جدید در الگوریتم ژنتیک در تحلیل رمز DES16 معرفی کردند. کنکایارو [۱۶] یک گزارش چندین ساله بر الگوریتم رمزنگاری DES ارائه داد. لاسکاری و همکاران [۱۷] روش هوش تکاملی را جهت تحلیل رمز متون به کار گرفتند. نالینی [۱۸] و همکاران تحلیل رمز SDES را با بهینه‌سازی اکتشافی انجام دادند. نما و همکاران [۱۹] یک گزارش از روش‌های رمزنگاری جهت امنیت داده‌ها ارائه دادند. تدرس و همکاران [۲۰] الگوریتم ژنتیک را جهت تحلیل رمز DES استفاده کردند. در این تحقیق، اساس کار بر بهینه کردن الگوریتم استاندارد بهینه‌سازی پرتو ذرات می‌باشد. با بهینه شدن، عملکرد این الگوریتم در مقایسه با الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک کارهای پیشین در معیار کاهش زمان کشف بیت‌های کلید اصلی، بهبود عملکرد داشته و در تمام قالب‌های متنی، قادر به کشف کامل بیت‌های کلید اصلی نموده و متن اصلی می‌تواند رمزگشایی شود.

[1] Symmetric Key Cryptosystem
[2] Asymmetric Key Cryptosystem
[3] Data Encryption Standard
[4] Simplified-Data Encryption Standard
[5] Brute Force Attack
[6] Standard Particle Swarm Optimization
[7] Memetic

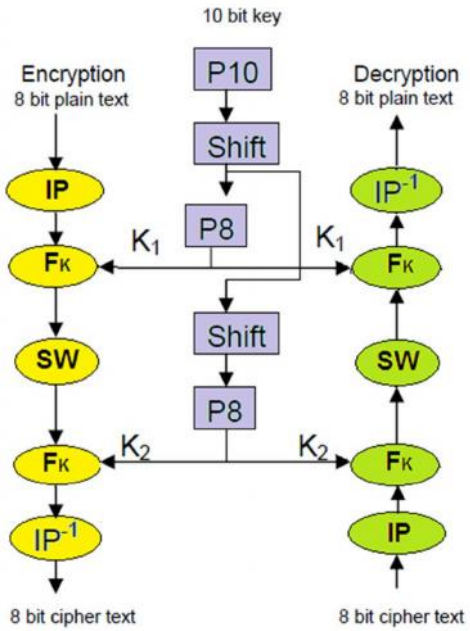
۳. الگوریتم رمزنگاری SDES

این الگوریتم نسخه ساده شده DES می‌باشد. در این الگوریتم هشت بیت داده اصلی با ده بیت کلید اصلی به عنوان ورودی، به هشت بیت داده رمز شده تبدیل می‌شود [۲۱]. این الگوریتم همان DES است، با این تفاوت که تعداد دورها از شانزده به دو دور و تعداد بیت‌های کلید اصلی از پنجاه و شش بیت به ده بیت کاهش یافته است. دو کلید فرعی هشت بیتی از ده بیت کلید اصلی استخراج می‌شود و هشت بیت داده اصلی، به هشت بیت داده رمز شده تبدیل می‌شود.

الگوریتم رمزگشایی SDES به صورت معکوس رمزگذاری می‌باشد [۲۱]. در زیر در مورد کلید اصلی رمزنگاری و رمزگشایی و همچنین توابع استفاده شده در این الگوریتم بحث شده است.

۳-۱. تولید کلید فرعی از کلید اصلی

در الگوریتم رمزنگاری SDES از کلید اصلی ده بیتی دو کلید فرعی هشت بیتی استخراج می‌شود که در ابتدا کلید اصلی به صورت P۱۰= [۳ ۵ ۲ ۷ ۴ ۱۰ ۱ ۹ ۸ ۶] جایگشت می‌شود، سپس یک شیفت عملیاتی به چپ انجام می‌شود، خروجی شیفت عملیاتی به صورت P۸= [۶ ۳ ۷ ۴ ۸ ۵ ۱۰ ۹] جایگشت (هشت بیتی) می‌شود و اولین کلید فرعی تولید می‌شود. خروجی شیفت عملیاتی در این مرحله یک بار دیگر شیفت عملیاتی به چپ روی آن انجام می‌شود (دو شیفت عملیاتی) و سپس جایگشت هشت بیتی مانند P۸ انجام شده و دومین کلید فرعی هم تولید می‌شود [۲۱]. شمای کلی الگوریتم رمزنگاری SDES در شکل (۱) آمده است.



شکل ۱. نمودار رمزنگاری SDES [۲۱]

۳-۲. فرایند رمزنگاری

فرایند رمزنگاری از مراحل زیر تشکیل شده است:

۱. جایگشت اولیه و نهایی[1] (IP): هشت بیت داده اصلی که به عنوان ورودی الگوریتم در نظرگرفته شده است به وسیله تابع IP = [۲ ۶ ۳ ۱ ۴ ۸ ۵ ۷] جایگشت داده می‌شود، در آخرین مرحله معکوس جایگشت به صورت IP^{-1} = [۴ ۱ ۳ ۵ ۷ ۲ ۸ ۶] روی داده اصلی اعمال می‌شود.

۲. تابع F_K[2]: این تابع، تابع پیچیده‌ای در الگوریتم رمزنگاری SDES می‌باشد که شامل ترکیبی از جایگشت و جانشینی تابع است [۲۱]. نیمه چپ و راست به صورت زیر مقداردهی می‌شود و تابع F در نیمه راست در بخش دوم رابطه (۱) آمده است [۱].

$$\begin{cases} L_{i-1} = R_{i-1} \\ R_{i-1} = L_{i-1} \oplus F(R_{i-1}, K_i) \end{cases} \quad (۱)$$

که در آن، L و R چهار بیت در سمت چپ و سمت راست ورودی می‌باشد. XOR، عملیات OR انحصاری است و Key هم کلید فرعی است [۱]. محاسبه F(R,Key) در زیر آمده است:

۱- اضافه کردن توسعه – جایگشت به صورت چهار بیت که با استفاده از توسعه E/P = [۴ ۱ ۲ ۳ ۲ ۳ ۴ ۱] انجام می‌شود.

۲- اضافه کردن هشت بیت کلیدفرعی و XOR کردن آن با خروجی مرحله ۱.

۳- چهار بیت سمت چپ در جعبه جایگشت S_0[3] و چهار بیت سمت راست در جعبه جایگشت S_1 قرار گرفته است.

۴- اضافه کردن جایگشت P_4 = [۲ ۴ ۳ ۱].

دو جعبه جایگشت تعریف شده به صورت جدول (۱) است.

جدول ۱. جعبه جایگشت مورد استفاده

S_0				S_1			
۱	۰	۳	۲	۰	۱	۲	۳
۳	۲	۱	۰	۲	۰	۱	۳
۰	۲	۱	۳	۳	۰	۱	۰
۳	۱	۳	۲	۲	۱	۰	۳

جعبه جایگشت به صورت زیر کار می‌کند:

اولین و چهارم بیت به عنوان سطر و دومین و سومین بیت به عنوان ستون جعبه جایگشت در نظر گرفته می‌شود و روی سطر و ستون در نهایت دو تا دو بیت خروجی به دست می‌آید (چهار بیت). تابع سوییچ[4] چهار بیت سمت چپ و راست را با هم عوض می‌کند و برای دور دوم هم باز توابع E/P، S_0، S_1 و P4 را به وسیله کلید فرعی

[1] Intial Permutation
[2] Function Key
[3] Spox
[4] Switch

دوم مشابه دور اول انجام می‌شود و الگوریتم رمزگذاری با دو دور خاتمه می‌یابد [۲۱].

۳-۳. الگوریتم رمزگشایی

رمزگشایی معکوس الگوریتم رمزگذاری است، هشت بیت داده رمز شده با ده بیت کلید اصلی به هشت بیت داده اصلی تبدیل می‌شود. بلوک رمزگشایی مثل بلوک رمزگذاری است، با این تفاوت که جای کلید فرعی یک و دو عوض می‌شود و تابع F_K معکوس می‌شود.

۳-۴. رمزنگاری و رمزگشایی متون

جهت رمزنگاری متن‌ها، از بخش رمزنگاری الگوریتم SDES برای هر حرف یک بلوک متنی استفاده شده است و جهت رمزگشایی متن‌ها، از بخش رمزگشایی الگوریتم SDES برای هرحرف یک بلوک متنی استفاده شده است.

۴. الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده

برای اولین بار ایده اصلی الگوریتم استاندارد بهینه‌سازی پرتو ذرات توسط کندی و ابرهارت به عنوان یک روش جستجوی غیر قطعی برای بهینه‌سازی تابعی مطرح گشت. این الگوریتم از حرکت دسته جمعی پرندگان که به دنبال غذا می‌گردند الهام گرفته شده است [۲۲]. در این تحقیق، الگوریتم استاندارد بهینه‌سازی پرتو ذرات، با تنظیم پارامترهای آن و با طراحی الگوریتمی جهت شکستن الگوریتم رمزنگاری SDES بهینه شده است. با توجه به ساختار الگوریتم، گروهی از کلیدها در یک جمعیت به صورت تصادفی دنبال کلید اصلی رمزنگاری SDES می‌گردند. هیچ یک از کلیدها محل کلید اصلی را نمی‌دانند، یکی از بهترین راه‌حل‌ها می‌تواند دنبال کردن کلیدی باشد که کمترین فاصله را تا کلید اصلی داشته باشد که این کمترین فاصله به وسیله برازندگی کلیدها به دست می‌آید. هر کلید دارای یک سرعت است که هدایت حرکت کلید را بر عهده دارد که هر کلید با دنبال‌کردن کلیدهای بهینه در حالت فعلی، به حرکت خود در فضای مسئله ادامه می‌دهد.

۴-۱. الگوریتم و روند جریان طراحی شده جهت شکستن الگوریتم رمزنگاری SDES

الگوریتمی که جهت شکستن الگوریتم رمزنگاری SDES به کار می‌رود به شرح زیر است:

۱- ورودی الگوریتم، متن رمز شده می‌باشد.

۲- تولید تصادفی جمعیت اولیه به عنوان کلید اصلی.

۳- ارزیابی هرکلید به وسیله معکوس الگوریتم رمزنگاری SDES و متن رمز شده، برای به دست آوردن متن اصلی.

۴- محاسبه برازندگی برای هر کلید با توجه به تابع هزینه.

۵- ارزیابی شرط که در صورت برقراری شرط (پایان دورها) الگوریتم پایان می‌یابد در غیر این صورت به گام شش می‌رود.

۶- دور جدید به صورت زیر به وجود می‌آید :

۶-۱ محاسبه ضریب‌وزنی هر کلید براساس رابطه (۲).

۶-۲ محاسبه بهترین موقعیتی که کلید تاکنون داشته است.

۶-۳ محاسبه بهترین موقعیتی که کل مجموعه کلیدها تاکنون داشته است.

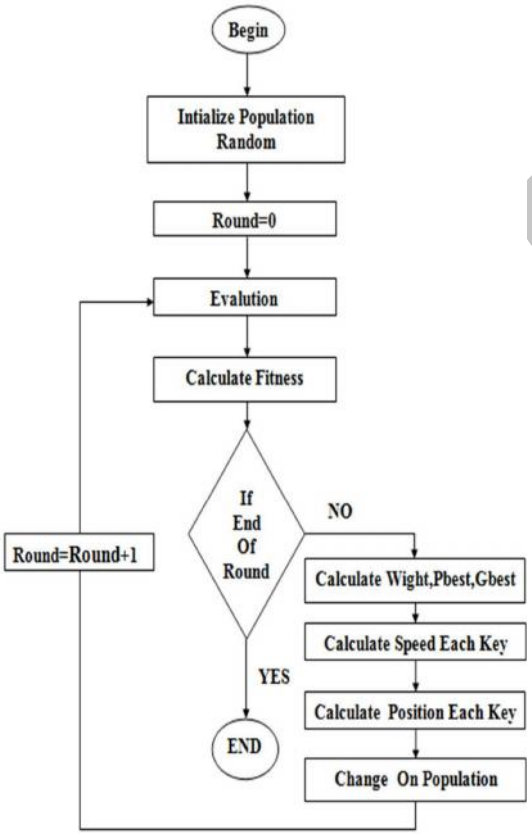
۶-۴ محاسبه سرعت هر کلید بر اساس رابطه (۳).

۶-۵ محاسبه مکان هر کلید بر اساس رابطه (۴).

۶-۶ همه تغییرات در جمعیت جدید اعمال می‌شود.

۶-۷ رفتن به مرحله ۳.

روند جریان طراحی شده برای الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده در شکل (۲) آمده است:



شکل۲ . روند جریان الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده

تنظیم پارامترها در خروجی الگوریتم در هر حوزه‌کاری تأثیرگذار می‌باشد. در این تحقیق پارامترها براساس شکستن الگوریتم رمزنگاری SDES تنظیم شده است.

۴-۲. تنظیم پارامترها

ضریب وزنی(w): وزن‌میانی می‌تواند یک ضریب ثابت، یک تابع خطی با زمان و یا حتی یک تابع غیرخطی با زمان نیز باشد. در این تحقیق بر اساس آزمون‌های انجام شده، وزن میانی یک تابع خطی با زمان در نظر گرفته شده است که این تابع خطی می‌تواند کاهش دهنده مشکل گیر افتادن در کمینه‌های محلی باشد. به این ترتیب در ابتدا، قسمت بیشتری از سرعت فعلی کلید در سرعت آینده‌اش دخیل می‌شود و با گذر زمان، این میزان کاهش می‌یابد. به عبارت بهتر، در ابتدا کلیدها میل بیشتری به حرکات انفجاری و تجربه‌های تازه دارند و با گذشت زمان با استفاده از ضریب وزنی (تابع خطی با زمان) در نظر گرفته شده، این میل جای خود را به دنباله‌روی بیشتر از بهترین‌ها می‌دهد. محاسبه ضریب وزنی هرکلید در رابطه (۲) آمده است.

بهترین موقعیتی که هر کلید تاکنون داشته است (P_{BSET}): در این تحقیق، P_{BSET} بر اساس برازندگی هر کلید در پیدا کردن بیت‌های داده اصلی، طراحی شده است، بدین صورت که بهترین موقعیت هر کلید بر اساس برازندگی به دست می‌آید و P_{BSET} را مقداردهی می‌کند. سپس در هر نسل بهترین موقعیتی هر کلید با نسل‌های قبل مقایسه شده، در صورتی که در نسل فعلی بهترین موقعیت هر کلید در مقایسه با نسل قبل بهتر شده باشد، P_{BSET} تغییر می‌کند در غیر این صورت P_{BSET} تغییر نمی‌کند و با P_{BSET} قبلی کار محاسبه سرعت و مکان کلید را ادامه می‌دهد.

بهترین موقعیتی که کل مجموعه کلیدها تاکنون داشته است (G_{BSET}): در این تحقیق G_{BSET} بر اساس برازندگی کل کلیدها در پیدا کردن بیت‌های داده اصلی، طراحی شده است، بدین صورت که بر اساس برازندگی انتخاب بهترین کلید در کل کلیدها، G_{BSET} را مقداردهی می‌کند. سپس در هر نسل بهترین موقعیتی کل کلیدها را با نسل‌های قبل مقایسه کرده، در صورتی که در نسل فعلی بهترین موقعیت کل کلیدها در مقایسه با نسل قبل بهتر شده باشد، G_{BSET} تغییر می‌کند در غیراین‌صورت G_{BSET} تغییر نمی‌کند و با G_{BSET} قبلی کار محاسبه سرعت و مکان کلید را ادامه می‌دهد.

محاسبه سرعت و مکان کلیدها: هر کلید با استفاده از رابطه‌های (۴-۲) به‌روز می‌شود [۲۲ و ۲۳].

$$W = \left(\left(W_{MAX} - \left(W_{MAX} - \frac{W_{MIN}}{ITR_{MAX}}\right)\right) * ITR\right) \quad (۲)$$

$$V[] = W * V[] + C_1 * RAND() * (P_{Best}[] - POSITION[]) + C_2 * RAND() * (G_{Best}[] - POSITION[]) \quad (۳)$$

$$POSITION[] = POSITION[] + V[] \quad (۴)$$

در رابطه (۲) ضریب وزنی محاسبه می‌شود که بر اساس آزمون‌های انجام شده W_{MAX}=۰/۹، W_{MIN} =۰/۴ ، ITR_{MAX}=۱۰ و ۱≤ITR≤۱۰ در نظرگرفته شده است که ITR تکرار دورهاست.

در رابطه‌های (۳ و ۴)، سرعت کلید و موقعیت محل فعلی کلید محاسبه می‌شود. Rand() یک عدد تصادفی در بازه صفر و یک است. C1 و C2 نیز پارامترهای یادگیری هستند و معمولاً در تمامی حوزه‌های کاری C1=C2=۲ در نظر گرفته شده است. P_{BSET} بهترین موقعیتی که هر کلید تاکنون داشته است که هر کلید بر اساس برازندگی در هر دور با کلیدهای قبل مقایسه شده و در صورت برازندگی بالا، به عنوان P_{BSET} انتخاب می‌شود. G_{BSET} بهترین موقعیتی که کل مجموعه کلیدها تاکنون داشته است. براساس برازندگی کل کلیدها در هر دور با کل کلیدها دورهای قبل مقایسه شده و انتخاب یک بهترین کلید در کل مجموعه کلیدها به‌عنوان G_{BSET} انتخاب می‌شود.

تابع‌هزینه[1]: رابطه (۵) یک تابع برازندگی عمومی برای تعیین کلید مناسب می‌باشد. در این رابطه زبان الفبایی برای زبان انگلیسی ([A-Z]) می‌باشد. K وD زبان آماری شناخته شده و پیام آماری رمزگشایی شده می‌باشد. a، b و t تک حرفی و دو حرفی و سه حرفی آماری هستند و α، β و ϒ ضریب وزنی تخصیص داده شده به هر سه حالت حرفی آماری می‌باشد که ϒ + β + α = ۱ می‌باشد [۲۲]. در بیشتر تحقیق‌ها به دلیل پیچیدگی حالت‌های مختلف حروف در رمزگشایی ازحالت تک حرفی استفاده شده است. در این تحقیق با استفاده از همین تابع‌هزینه از هر سه حالت حرفی استفاده شده است.

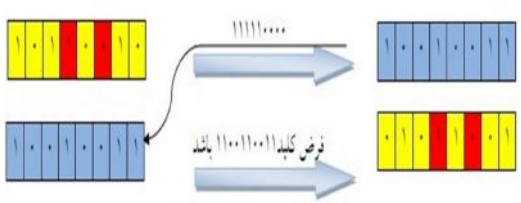
$$C_K = \alpha \Sigma(i \varepsilon \tilde{A}) |k(i)u - D(i)u| + \beta \Sigma(i,j \varepsilon \tilde{A}) |k(i,j)b - D(i,j)b| + \gamma \Sigma(i,j,k \varepsilon \tilde{A}) |k(i,j,k)t - D(i,j,k)t| \quad (۵)$$

جمعیت[2]: مفهوم جمعیت در الگوریتم رمزنگاری، جمعیتی از کلیدهای اصلی می‌باشد. برای مسئله کلیدهایی وجود دارند که می‌توانند به عنوان کلید، چه درست چه غلط در نظر گرفته شوند. در این تحقیق بر اساس آزمون‌های انجام شده با در نظر گرفتن معیار کشف بیت‌های کلید اصلی و زمان کشف بیت‌های کلید اصلی، جمعیت کلیدها ۸۰ درنظر گرفته شده است.

محاسبه برازندگی[3]: محاسبه برازندگی کلید، از اِعمال تبدیل مناسب بر روی تابع هدف یعنی تابعی که قرار است بهینه شود به دست می‌آید. در شکستن الگوریتم رمزنگاری SDES، محاسبه برازندگی هر کلید بر اساس رمزگشایی متن رمز شده و به دست آوردن متن اصلی طراحی شده است. به عنوان مثال در شکل (۳) حرف THE با کلید ۱۱۱۱۱۰۰۰۰۰ به حرف ٪ رمز می‌شود، سپس هر کلیدی را با حرف ٪ که معادل با هشت بیت به عنوان ورودی

[1] Cost Function
[2] Population
[3] Fitness Calculation

است رمز می‌شود و یک حرف با هشت بیت به عنوان خروجی به دست می‌آید. سپس تعداد بیت‌های آن حرف با هشت بیت به عنوان خروجی را باحرف THE با هشت بیت به عنوان ورودی آزمون می‌شود، تعداد مکان‌های که دارای بیت‌هایی یکسان هستند به عنوان برازندگی محاسبه می‌شود. در شکل (۳) برازندگی عدد ۲ است.



شکل ۳. محاسبه برازندگی

۵. نتایج و بحث

در این تحقیق، سیستم core 2 duo باپردازنده ۲.۵۳ گیگا هرتز و حافظه RAM چهار گیگابایت استفاده شده است. سیستم‌های مورد استفاده در کارهای پیشین که جهت ارزیابی و مقایسه‌ها استفاده شده است، core 2 duo بوده است [۲]. از بلوک‌های مختلف متنی جهت شکستن الگوریتم رمزنگاری SDES استفاده شده و در تابع هزینه از خصوصیات آماری تک حرفی و دو حرفی و حتی سه حرفی استفاده شده است که در کارهای پیشین تابع هزینه از خصوصیات آماری تک حرفی استفاده شده است. در شکستن الگوریتم رمزنگاری SDES با توجه به ماهیت الگوریتم‌های فرااکتشافی میانگینی از اجراهای متفاوت، به عنوان نتایج تحقیق در نظر گرفته شده است. پارامترهای استفاده شده در الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده در جدول (۲) آمده است.

جدول ۲. تنظیم پارامترها

پارامترها	مقدار
پارامتر یادگیری C1	۲
پارامتر یادگیری C2	۲
ضریب وزنی	۱۰≤تکرار دورها≤۱
جمعیت اولیه	۸۰
تعداد تکرار	۱۰

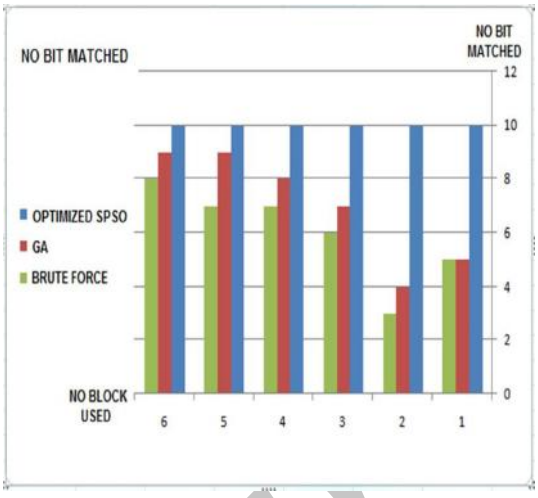
پارامترهایی که برای الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده در نظر گرفته شد، بهترین مقادیری است که تحقیق شده است.

در این تحقیق، هدف کشف بیت‌های کلید اصلی در کمترین زمان جهت رمزگشایی و شکستن الگوریتم رمزنگاری SDES در تمامی بلوک‌های متنی می‌باشد. درالگوریتم جستجوی حمله قوی با آزمودن تمامی کلیدهای ممکن سعی می‌شود که کلید واقعی که متن رمز شده را به متن اصلی تبدیل می‌کند را پیدا کند ولی برای حدس زدن کلید واقعی زمان زیادی صرف می‌شود [۲] و تمامی بیت‌های کلید اصلی کشف نمی‌شود. در الگوریتم ژنتیک [۲] به کار گرفته شده در کارهای پیشین، زمان کشف بیت‌های کلید اصلی بیش از یک دقیقه بوده و کشف کامل بیت‌های کلیدرمز در هیچ بلوکی متنی امکان‌پذیر نبوده است (عدم رمزگشایی). در این تحقیق جهت شکستن الگوریتم رمزنگاری SDES، الگوریتم استاندارد بهینه‌سازی پرتو ذرات استفاده شده است. جهت شکستن الگوریتم رمزنگاری SDES، الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده است، با بهینه شدن الگوریتم استاندارد بهینه‌سازی پرتو ذرات درمقایسه با کارهای پیشین، زمان کشف بیت‌های کلید اصلی کاهش یافته و در تمام بلوک‌های متنی قادر به کشف کامل بیت‌های کلید اصلی نموده که می‌توان متن‌اصلی را در تمام بلوک‌ها رمزگشایی کرد. بلوک های متنی متفاوتی در هر سه الگوریتم (الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک) تحقیق شده است که نتایج در دو معیار ارزیابی زمان به ثانیه و کشف بیت‌های کلید اصلی بررسی شده است. نتایج نشان می‌دهند که الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، ضمن کشف کامل بیت‌های کلیداصلی در کوتاه‌ترین زمان جهت رمزگشایی، به شکسته شدن الگوریتم رمزنگاری SDES منجر شده است. نتایج آماری کشف بیت‌های کلیداصلی در الگوریتم رمزنگاری SDES در جدول (۳) آمده است.

جدول ۳. تعداد بیت‌های کشف شده در کلیداصلی

بلوک متنی	تعداد حروف در بلوک متنی	حمله قوی (تعداد بیت‌های کشف شده)	الگوریتم ژنتیک (تعداد بیت‌های کشف شده)	استاندارد بهینه سازی پرتو ذرات بهینه شده (تعداد بیت‌های کشف شده)
۱	۲۰۰	۵	۵	۱۰
۲	۴۰۰	۳	۴	۱۰
۳	۶۰۰	۶	۷	۱۰
۴	۸۰۰	۷	۸	۱۰
۵	۱۰۰۰	۷	۹	۱۰
۶	۱۲۰۰	۸	۹	۱۰

در جدول (۳)، در الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک کارهای پیشین [۲]، تمامی بیت‌های کلید اصلی جهت رمزگشایی کشف نشده است (در تمامی بلوک‌ها) در صورتی که با استفاده از الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، بیت‌های کلید اصلی در تمامی بلوک‌ها به صورت کامل کشف شده است. با کشف کامل بیت‌های کلید اصلی در تمامی بلوک‌ها، متن اصلی رمزگشایی می‌شود و الگوریتم رمزنگاری SDES شکسته می‌شود. نمودار کشف بیت‌های کلید اصلی در شکل (۴) آمده است.



شکل ۴. تعداد بیت‌های کشف شده در کلید اصلی

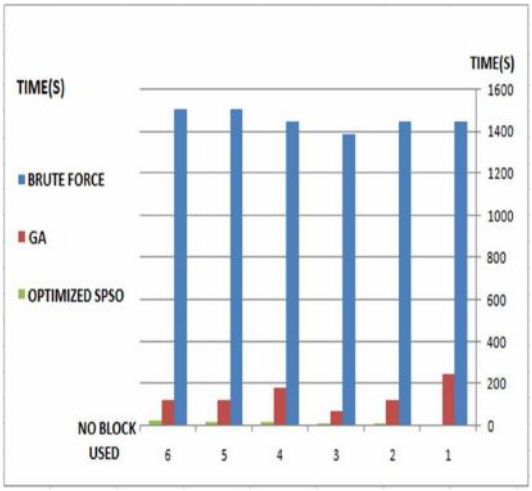
الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک در شکل (۴) استفاده شده است. محور افقی تعداد بلوک مورد استفاده و محور عمودی تعداد بیت کشف شده در کلید اصلی می‌باشد که همان‌طور که در شکل (۴) دیده می‌شود الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده در تمامی بلوک‌ها، بیت‌های کلید اصلی را کشف کرده است در حالی که در الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک در هیچ بلوک متنی تمای بیت‌های کلید اصلی کشف نشده است. نتایج آماری زمان کشف بیت‌های کلید اصلی در جدول (۴) آمده است.

جدول ۴. زمان‌کشف بیت‌های کلید اصلی

بلوک متنی	تعداد حروف در بلوک متنی	حمله قوی (ثانیه)	الگوریتم ژنتیک (ثانیه)	استاندارد بهینه سازی پرتو ذرات بهینه شده (ثانیه)
۱	۲۰۰	۱۴۴۳	۲۴۷	۳
۲	۴۰۰	۱۴۴۷	۱۲۱	۵
۳	۶۰۰	۱۳۸۶	۶۹	۷
۴	۸۰۰	۱۴۴۱	۱۸۱	۹
۵	۱۰۰۰	۱۵۰۱	۱۲۶	۱۱
۶	۱۲۰۰	۱۵۰۵	۱۲۱	۱۳

در جدول (۴)، زمان کشف بیت‌های کلید اصلی در الگوریتم جستجوی حمله‌قوی و الگوریتم ژنتیک کارهای پیشین [۲] بالاست (بیش از یک دقیقه) در حالی که با استفاده از الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، زمان کشف بیت‌های کلید اصلی کاهش یافته است (کمتر از بیست ثانیه). نمودار زمان کشف بیت‌های کلید اصلی در شکل (۵) آمده است.

الگوریتم جستجوی حمله قوی، الگوریتم ژنتیک و الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده در شکل (۵) استفاده شده است. محور افقی تعداد بلوک مورد استفاده و محور عمودی زمان کشف بیت‌های کلید اصلی می‌باشد که در الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده کاهش زمان محسوسی دیده می‌شود.



شکل ۵. زمان کشف بیت‌های کلید اصلی

۶. نتیجه‌گیری

در این تحقیق جهت شکستن رمز الگوریتم رمزنگاری SDES، الگوریتم فرااکتشافی استاندارد بهینه‌سازی پرتو ذرات استفاده شده است. الگوریتم استاندارد بهینه‌سازی پرتو ذرات با تنظیم پارامترهای آن و طراحی الگوریتمی جهت شکستن الگوریتم رمزنگاری SDES بهینه شده است. نتایج نشان می‌دهد که الگوریتم استاندارد بهینه‌سازی پرتو ذرات بهینه شده، عملکرد بهتری در شکستن الگوریتم رمزنگاری SDES از خود نشان داده است به طوری که در مقایسه با الگوریتم جستجوی حمله قوی و الگوریتم ژنتیک کارهای پیشین در معیار کشف بیت‌های کلید اصلی، ده بیت کلید اصلی کشف شده است و در معیار زمان کشف بیت‌های کلید اصلی، زمان از بیش یک دقیقه به کمتر از بیست ثانیه کاهش یافته است که با کشف تمامی بیت‌های کلید اصلی منجر به شکسته شدن الگوریتم رمزنگاری SDES شده است. در آینده با این روش جدید، شکست الگوریتم‌های مدرن DES، 3DES و AES بررسی خواهد شد.

۷. مراجع

[1] Zakerolhossini, A.; Malekian, E. "Security Data"; Tehran: Scientific and Cultural Institute Nas, 1390; 1 , 11-239.

[2] Sharma, L.; Pathak, B. K.; Sharma, R. "Breaking of Data Simplified Encryption Standard Using Genetic Algorithm"; Global J. of Computer Science and Technology, 2012, 12, 55-60.

[3] Abd-Elmonim, W.G.; Ghali, N. I.; Hassanien, A. E.; Abraham, A. "Known-Plaintext Attack of DES-16 Using Particle Swarm Optimization"; Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on IEEE, 2011, 12 - 16.

[4] Akiwate, B.; Desai, V. "Artificial Neural Networks For Cryptanalysis Of DES"; International Journal of Innovations in Engineering and Technology, 2013, 2, 11-17.

[5] Alallayah, K. M.; El-Wahed, W. F. A.; Amin, M.; Alhamami, A. H. "Attack of Against Simplified Data Encryption Standard Ciphersystem Using Neural Networks"; J. Computer Sci. 2010, 1, 29-35.

[6] Suman, G.; Krishna, C. "Improved Cryptosystem Using SDES Algorithm With Substitution Ciphers"; Int. J. of Advanced Research in Computer Sci. and Software Eng., 2013, 3, 131-136.

[7] Al-Shakarchy, N. D. K. "Simulating Des Algorithm Using Artificial Neural Network"; J. of Kerbala Univ., 2012, 10, 13-21.

[8] Garg, P. "A Comparison Between Memetic Algorithm and Genetic Algorithm for the of Cryptanalysis Simplified Data Encryption Standard Algorithm"; Int. J. of Network Security & Its Applications, 2009, 1, 34-42.

[9] Alanazi Hamdan O.; Zaidan, B. B.; Zaidan, A. A.; Jalab, H. A.; Shabbir, M.; Al-Nabhani, Y. "New Comparative Study Between DES, 3DES and AES Within Nine Factors"; J. of Computing, 2010, 2, 152-157.

[10] Husein, H. M. H.; Bayoumi, B. I.; Holail, F. S. B.; Hasan, E. M.; El-Mageed, M. Z. A. "A Genetic Algorithm for Cryptanalysis of DES-8"; Int. J. of Network Security, 2007, 5, 213–219.

[11] Sharma, I. R. "Comparative Analysis of DES and SDES Encryption Algorithm Using Verilog Coding"; Int. J. of Innovative Research In Electrical, 2013, 1, 469-473.

[12] Jadon, S. S.; Sharma, H.; Kumar, E.; Bansal, J. C. "Application of Binary Particle Swarm Optimization in Cryptanalysis of DES"; Advances in Intelligent and Soft Computing, 2012, 130, 1061–107.

[13] Kendhe, A. K.; Agrawal, H. "A Survey Report on Various Cryptanalysis Techniques"; Int. J. of Soft Computing and Engineering, 2013, 3, 287-293.

[14] Salabat, K.; Armughan, A.; Yahya, D. M. "Ant-Crypto, a Cryptographer for Data Encryption Standard"; Int. J. of Computer Sci. Issues, 2013, 10, 400-406.

[15] Sathya, S. S.; Chithralekha, T.; Anandakumar, P. "Nomadic Genetic Algorithm for Cryptanalysis of DES 16"; Int. J. of Computer Theory and Engineering, 2010, 2, 411-415.

[16] Kenekayoro, P. T. "The Data Encryption Standard Thirty Four Years Later: An Overview"; African J. Mathematics and Computer Sceince Research, 2010, 3, 267-269.

[17] Laskari, E. C.; Meletiou, G. C.; Stamatiou, Y. C.; Vrahatis M. N. "Applying Evolutionary Computation Methods for the Cryptanalysis of Feistel Ciphers"; Applied Mathematics and Computation, 2007, 184, 63-72.

[18] Nalini, N.; Raghavendra, G. R. "Cryptanalysis of Simplified Data Encryption Standard via Optimisation Heuristics"; Int. J. of 240 Computer Science and Network Security, 2006, 6, 240-246.

[19] Nema, P.; Jain, P. A. "A Comparative Survey on Various Encryption Techniques for Information Security"; Int. J. of Advanced Research in Computer Science and Software Engineering, 2013, 3, 725-730.

[20] Tadros, T.; Hegazy, A. E. F.; Badr, A. "Genetic Algorithm for DES Cryptanalysis"; Int. of J. of computer and Network Security, 2010, 10, 5-11.

[21] Garg, P. "Cryptanalysis of SDES via Evolutionary Computation Techniques"; Int. J. of Computer Science and Information Security, 2009, 1.

[22] Sharma, L.; Pathak, B. K; Sharma, N. "Breaking of Simplified Data Encryption Standard Using Binary Particle Swarm Optimization"; Int. J. of Computer Science Issues, 2012, 9, 307-313.

[23] Bansal, J. C.; Singh, P. K.; Saraswat, M.; Verma, A.; Jadon, S. S.; Abraham, A. "Inertia Weight Strategies in Particle Swarm Optimization"; Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on IEEE, 2011, 633-640.