دومین کنفرانس بین‌المللی ترکیبیات، رمزنگاری و محاسبات
I4C 2017
Homepage: http://i4c.iust.ac.ir

فازی سازی یک الگوریتم رمزنگاری جدید

رضا بیات تاجور*[۱] ، محمدصادق کیانی[۲]

۱-کارشناس ارشد ریاضی، دانشگاه پدافند هوایی خاتم الانبیاء(ص)

۲- کارشناس ارشد مهندسی صنایع، دانشگاه پدافند هوایی خاتم الانبیاء(ص)

چکیده

ارسال اطلاعات از یک نقطه به نقطه دیگر، تبادل اطلاعات نامیده می‌شود. امروزه امنیت یک موضوع مهم در تبادل اطلاعات می‌باشد. یکی از مهم‌ترین ابزارهایی که برای تامین امنیت به‌کار می‌رود، علم رمزنگاری می‌باشد. الگوریتم رمزنگاری یک روش ریاضی برای رمزنگاری داده‌ها می‌باشد. ما در این مقاله الگوریتمی را معرفی می‌کنیم که دارای سطح پردازش پایین و سطح امنیت بالا مطابق با نیاز کاربر بوده و همچنین از الگوریتم‌های رمزنگاری موجود، پیشرفته‌تر باشد. برای اجرای این الگوریتم از منطق فازی استفاده می‌کنیم. در انتها نتایج الگوریتم پیشنهادی را با دیگر الگوریتم‌های رمز-نگاری مقایسه کرده و مورد تجزیه و تحلیل قرار خواهیم داد.

کلمات کلیدی: الگوریتم، امنیت، تبادل اطلاعات، رمزنگاری، رمزگشایی، منطق فازی

۱. مقدمه

رمزنگاری از ریشه یونانی کلمات *Kryptos* و *Graphin* گرفته شده است. تاریخچه استفاده از رمزنگاری به قرن‌ها قبل باز می‌گردد. به‌طوری که در گذشته نیز به منظور محافظت از پیغام‌هایی که بین فرماندهان، جاسوسان، دیپلمات‌ها و... رد و بدل می‌شده از رمزنگاری استفاده می‌کردند. رمزشناسی علمی است که در رابطه با موضوعات رمزنگاری و رمزگشایی بحث می‌کند. رمزنگاری فرآیندی است که یک پیام قابل خواندن را به یک فرمت غیرقابل خواندن تبدیل می‌کند. رمزگشایی به فعالیت‌های مرتبط با تجزیه و تحلیل رمز پرداخته و به دنبال دستیابی به کلید رمز و در نتیجه به‌دست آوردن متن اصلی از روی متن رمز می‌باشد [۵]. دانش رمزنگاری یکی از معدود شاخه‌های دانش معاصر است که هویت خود را از ناتوانی بشر در حل مسائل ریاضی کسب می‌کند. متدهای رمزنگاری و رمزگشایی، همیشه یک عامل مهم در استراتژی‌های جنگی بوده‌اند. تلاش‌های گسترده آمریکائی‌ها برای شکستن رمز ژاپنی‌ها در طول جنگ جهانی دوم، یکی از مهمترین وظایف متفقین به شمار می‌آمد.

[1] **Corresponding author:** کارشناس ارشد ریاضی، دانشگاه پدافند هوایی خاتم الانبیاء(ص)
Email: r.bayat.tajvar@gmail.com

گفتنی است امنیت، مساله‌ای اساسی و اصلی در ارتباطات مدرن می‌باشد. بسیاری از جرائم سایبری با پیشرفت تکنولوژی به‌وجود می‌آیند. برای جلوگیری از خطرات امنیتی می‌توان کارهای مختلفی انجام داد. از قبیل:

- خاموش کردن سرویس‌های غیر مورد نیاز
- به روز کردن سیستم‌ها
- کاهش حق دسترسی به برنامه‌ها وکاهش تعداد کاربران
- استفاده از پروتکل‌های امنیتی

اما یک راه حل دیگر برای مقابله با خطرات امنیتی استفاده از رمزنگاری می‌باشد. رمزنگاری دو جزء اصلی دارد؛ الگوریتم و کلید. الگوریتم یک مبدل یا فرمول ریاضی است. مجموعه‌ای از قوانین که در یک فرآیند رمزنگاری استفاده می‌شود الگوریتم رمزنگاری نامیده می‌شود. یک کلید برای تبدیل اطلاعات اصلی استفاده می‌شود. این اطلاعات، غیرقابل خواندن و به عنوان متن پیام رمز شناخته می‌شوند. از آن جایی که امینت، یک نگرانی جدی در تبادل اطلاعات می‌باشد لذا الگوریتم-های رمزنگاری، یک بخش مهم از آن می‌باشد [۱]. اکثر الگوریتم‌های رمزنگاری که امروزه مورد استفاده قرار می‌گیرند، تنها نگران امنیت هستند. لیکن نحوه عملکرد نیز برای فن‌آوری‌های در حال پیشرفت بسیار مهم و ضروری است. الگوریتم‌های رایج امروزی، هم زمان پردازش بالایی دارند و هم از امنیت کافی برخوردار نیستند. اما کاربرانی که از یک پهنای باند کم استفاده می‌کنند نیاز به الگوریتمی دارند که قدرت پردازش پائینی داشته باشد. الگوریتم‌هایی که سطح امنیتی بالایی دارند، از قدرت پردازش بیشتری نسبت به الگوریتم‌های با سطح امنیتی پایین، برخوردار هستند. بعد از انجام مطالعات مقدماتی روی الگوریتم‌های موجود، متوجه می‌شویم که می‌بایست یک الگوریتم بهتر برای تامین امنیت کاربران ارائه شود. بنابراین ما در این مقاله روی موضوع امنیت تمرکز خواهیم کرد و می‌خواهیم سطح امنیتی را در تبادل داده، با استفاده از یک الگوریتم رمزنگاری جدید افزایش دهیم. ما این کار را با به‌کار بردن منطق فازی در یک الگوریتم رمزنگاری پیشرفته انجام خواهیم داد. در این الگوریتم پیشنهادی، کاربران با توجه به نیاز خود، طول کلید مورد نظر را انتخاب خواهند کرد. به‌طور کلی دو نوع الگوریتم وجود دارد:

- الگوریتم های کلید رمز متقارن
- الگوریتم های کلید رمز نامتقارن

در الگوریتم‌های کلید رمز متقارن، برای رمزگذاری و رمزگشایی پیام از یک کلید رمز استفاده می‌شود. الگوریتم‌های کلید رمز متقارن گاهی الگوریتم‌های کلید رمز محرمانه و گاهی هم الگوریتم‌های کلید رمز خصوصی نامیده می‌شوند. اما در الگوریتم‌های کلید رمز نامتقارن، یک کلید برای رمزگذاری پیام به‌کار می‌رود و کلید دیگر برای رمزگشایی آن. رمزنگاری کلید رمز عمومی، یک دسته مهمی از الگوریتم‌های کلید رمز نامتقارن است. در این الگوریتم‌ها معمولا کلید رمزگذاری را کلید رمز عمومی می‌نامند، چون می‌تواند بدون اینکه خدشه‌ای به محرمانه بودن پیام یا کلید رمزگشایی وارد کند در دسترس همگان قرار داشته باشد. کلید رمزگشایی پیام نیز معمولا کلید رمز خصوصی یا کلید رمز محرمانه نامیده می‌شود.

۲. الگوریتم DES

الگوریتم رمزنگاری، به هر الگوریتم یا تابع ریاضی گفته می‌شود که در فرایند رمزنگاری/ رمزگشایی مورد استفاده قرار گیرد. یکی از مهمترین الگوریتم‌های متقارن، الگوریتم DES می‌باشد. این الگوریتم در دهه ۷۰ میلادی در آمریکا به‌عنوان یک استاندارد کدگذاری مطرح شد. این الگوریتم این‌گونه عمل می‌کند که رشته‌ای از متن اصلی با طول ثابت را به عنوان ورودی می‌گیرد و پس از انجام یک سری اعمال پیچیده روی آن خروجی را که طولی برابر طول ورودی دارد تولید می‌کند. همچنینDES از یک کلید برای ایجاد رمز استفاده می‌کند و تنها کسانی قادر به رمزگشایی خواهند بود که مقدار کلید را می‌دانند. در DES طول قطعات ۶۴ بیت است. کلید نیز شامل ۶۴ بیت است ولی در عمل تنها از ۵۶ بیت آن استفاده می‌شود و از ۸ بیت دیگر فقط برای چک کردن Parity استفاده می‌شود. الگوریتم شامل ۱۶ مرحله مشابه است که هر مرحله یک دور نامیده می‌شود. متنی که قرار است رمزگذاری شود ابتدا در معرض یک جایگشت اولیه (IP) قرار می‌گیرد. سپس یک سری اعمال پیچیده وابسته به کلید روی آن انجام می‌شود و در نهایت در معرض یک جایگشت نهایی (FP) قرار می‌گیرد. IP و FP معکوس یکدیگرند. FP عملی که توسط IP انجام شده‌است را خنثی می‌کند؛ بنابراین از جنبه رمزنگاری اهمیت چندانی ندارند. قبل از دور اصلی، داده به دو بخش ۳۲ بیتی تقسیم می‌شود که این دو نیمه به طور متناوب مورد پردازش قرار می‌گیرند این تقاطع به عنوان شکل فیستل شناخته می‌شود. ساختار فیستل تضمین می‌کند که رمزگذاری و رمزگشایی دو رویه کاملاً مشابه هم هستند و تنها تفاوت آنها این است که زیرکلیدها در زمان رمزگشایی در جهت معکوس رمزگذاری به کار برده می‌شوند. تابعی که خروجی IP را می‌گیرد و پس از شانزده مرحله ورودی FP را فراهم می‌کند تابع F نامیده می‌شود. این تابع یک ورودی ۳۲ بیتی و یک ورودی ۴۸ بیتی دارد و یک خروجی ۳۲ بیتی تولید می‌کند. بلوک ورودی شامل ۳۲ بیت که نیمه سمت چپ را تشکیل می‌دهد و با L نشان داده می‌شود و به دنبال آن ۳۲ بیت دیگر که نیمه راست را تشکیل می‌دهد و با R نمایش داده می‌شود است. پس کل بلوک را می‌توان به صورت LR نمایش داد [۱].

۳. تحلیل الگوریتم پیشنهادی

در این بخش ابتدا الگوریتم پیشنهادی را در دو بخش رمزنگاری و رمزگشایی مورد تجزیه و تحلیل قرار داده، سپس با استفاده از فازی کردن الگوریتم، قوانین مربوطه را به‌دست می‌آوریم.

۳-۱ الگوریتم تولید کلید

فرض می‌کنیم که در ابتدا طول کلید ۶۴ بیت باشد. از این تعداد، ۸ بیت را برداشته لذا طول کلید ۵۶ بیت می‌شود. سپس کلید از طریق یک جایگشت تحت تاثیر قرار می‌گیرد و به دو نیمه ۲۸ بیتی تقسیم می‌شود. آنگاه هر دو نیمه شیفت داده شده و تحت تاثیر S-box قرار می‌گیرند. سرانجام نیمه چپ و راست با یکدیگر مرتبط شده و یک کلید را می‌سازند. ۳۲ زیرکلید مشابه همین روش برای یک فرایند رمزنگاری/ رمزگشایی تولید می‌شوند.

۳-۲ اجرای الگوریتم رمزنگاری

فرایند رمزنگاری در ۱۶ مرحله و با استفاده از ۳۲ زیرکلید انجام می‌شود. همان طور که در شکل (۲) نشان داده شده، دیتای اصلی را به دو بخش ۶۴ بیتی تقسیم می‌کنیم. هر دو نیمه راست و چپ با استفاده از تابع و کلید، رمزنگاری می-

شوند. ابتدا پیام ورودی، به سیستم باینری تبدیل شده و تحت تاثیر یک جایگشت اولیه قرار می‌گیرد. در جایگشت اولیه، پیام باینری شده، به دو بلوک ۳۲ بیتی تقسیم می‌شود. همه این بلوک‌ها مطابق با هشت الگو مرتب می‌شوند. سپس این بلوک‌ها در یک آرایه سه بعدی گذاشته می‌شوند. در مرحله اول از رمزنگاری، دو نیمه ۳۲ بیتی به همراه کلیدهای مربوطه XOR می‌شوند. سپس خروجی یک نیمه با خروجی نیمه دیگر XOR می‌شود. از این روش برای به دست آوردن مقدار نیمه چپ جدید استفاده می‌شود. خروجی XOR نیمه چپ با کلید مربوطه به عنوان مقدار نیمه راست جدید به دست می‌آید. پس از انجام همین فرآیند برای ۱۶ بار، می‌توانیم L16 و R16 را دریافت کنیم. سپس خروجی ۶۴ بیتی را از ترکیب نیمه L16 و R16 به دست آورده و در یک ماتریس قرار می‌دهیم. حال بایستی ماتریس جدید را در یک ماتریس ثابت ضرب کنیم. در نهایت خروجی را از طریق جدول IP معکوس، ارسال کرده و متن رمز را دریافت می‌کنیم.

۳-۳ اجرای الگوریتم رمزگشایی

در فرایند رمزگشایی متن رمز شده از طریق جدول IP معکوس، ارسال می‌شود. سپس در یک ماتریس ثابت ضرب شده و خروجی ۶۴ بیتی اصلی را به‌دست می‌آورد. در نهایت خروجی به دو نیمه ۳۲ بیتی تقسیم می‌شود. پس از اجرای این روش معکوس از فرآیند رمزنگاری می‌توانیم پیام اصلی را به دست آوریم.

۳-۴ فازی کردن الگوریتم

منطق فازی یک روش حل مسئله در سیستم کنترل است که در سیستم‌های مختلف اعم از ساده، کوچک، میکروکنترلرها، شبکه‌ها، رایانه‌های و سیستم‌های مبتنی بر جمع‌آوری داده مورد استفاده قرار می‌گیرد [۷]و[۸]. منطق فازی یک روش آسان برای رسیدن به یک نتیجه قطعی بر اساس اطلاعات مبهم، نادرست، غیردقیق و یا مخدوش می‌باشد. مفهوم یک زیر مجموعه فازی از یک مجموعه ناتهی برای اولین بار توسط پروفسور زاده (۱۹۶۵) معرفی شد [۳].

برای دستیابی به امنیت پایین و پردازش کم‌تر، الگوریتم از کلیدهای مختلف استفاده می‌کند. موقعیت "۰" به یک الگوریتم با پردازش پایین و موقعیت "۱" به یک الگوریتم کاملاً امن اختصاص داده می‌شود. فازی‌سازی با توجه به طول کلید و تعداد جدول‌های نگاشتی از یک الگوریتم رمزنگاری تغییر می‌کند. کاربران می‌توانند طول کلید مورد نظر را وارد کنند. طول یک کاراکتر ۸ بیت می‌باشد. ساختار الگوریتم اصلی، طول‌های مختلف کلید را تا ۱۲۸ بیت، تعریف می‌کند. کاربر می‌تواند کلید مورد نظر را وارد کرده و الگوریتم هم می‌تواند با توجه به تعداد جدول‌های نگاشتی (Tables Mapping) وزن مربوطه را اختصاص دهد. توزیع وزن‌ها از محدوده ۰ تا ۱ و تعداد سطوح امنیتی از ۱ تا ۱۶ متفاوت خواهد بود. تعداد مراحل به وسیله جدول‌های از پیش تعریف شده و ورودی اولیه کاربران، تعیین خواهد شد. جدول‌های نگاشتی در الگوریتم که از پیش تعریف شده‌اند شامل مقادیر معینی می‌باشند که این مقادیر به صورت دینامیکی الگوریتم مربوطه را انتخاب می‌کنند.

بهترین حالت آن است که طول کلید ورودی توسط کاربر، ۱۶ بیت و تعداد جدول‌های نگاشتی بین ۱ تا ۸ باشد. فرض کنیم که کاربر گزینه رمزنگاری با تعداد ۸ جدول نگاشتی را انتخاب کند. الگوریتم وزن بیشتری را به ورودی‌های داده شده توسط کاربر اختصاص می‌دهد. زیرا طول اولیه کلید ۱۶ بیت است که منجر به ایجاد بیشترین تعداد مراحل یعنی ۱۶ می‌شود. در نهایت وارد کردن بیشترین مقدار طول کلید باعث ایجاد وزن بیشتر و در نتیجه ایجاد بالاترین سطح از مراحل و بالاترین سطح امنیت خواهد شد.

بدترین حالت آن است که طول کلید ورودی توسط کاربر، ۱ بیت و تعداد جدول‌های نگاشتی بین ۱ تا ۸ باشد. فرض کنید که کاربر گزینه رمزنگاری با تعداد ۱ جدول نگاشتی را انتخاب کند. الگوریتم یک وزن کمتر را به ورودی‌های داده شده توسط کاربر، اختصاص می‌دهد. چون طول اولیه کلید ۱ بیت بوده و این پایین‌ترین مقدار ورودی کلید است لذا این باعث ایجاد یک وزن کمتر خواهد شد. بنابراین پایین‌ترین سطح امنیتی ایجاد خواهد شد.

اما حالت نرمال آن است که طول کلید ورودی توسط کاربر، ۸ بیت و تعداد جدول‌های نگاشتی بین ۱ تا ۸ باشد. فرض کنید که کاربر گزینه رمزنگاری با تعداد ۴ جدول نگاشتی را انتخاب کند. بنابراین الگوریتم وزن بیشتری را نسبت به بدترین حالت و وزن کمتری را نسبت به بهترین حالت اختصاص می‌دهد. وقتی که طول اولیه کلید مقدار متوسط یعنی ۸ بیت باشد، اندازه وزن نیز متوسط خواهد شد. با توجه به تعداد مراحل امنیتی (در این حالت ۸ می‌باشد)، سطح پایین امنیتی نیز در الگوریتم اجرا خواهد شد. برخی از قوانین منطق فازی به شرح جدول-۱ می‌باشند:

جدول ۱- برخی از قوانین منطق فازی

طول کلید (بیت)	سطح (لایه)	تعداد دور
۱	۰	۰
۲	۰	۱
۳	۰	۱
۴	۰	۲
۵	۰	۲
۶	۰	۳
۷	۰	۳
۸	۰	۴
۹	۰	۴
۱۰	۰	۵
۱۱	۰	۵
۱۲	۰	۶
۱۳	۰	۶

۴. پیاده سازی نتایج آزمون

بعد از اجرای الگوریتم رمزنگاری، نتایج به‌وسیله مقایسه با الگوریتم DES تست می‌شوند.

۴-۱ نتایج آزمون با تغییر اندازه فایل:

جدول ۲- مقایسه با توجه به اندازه فایل

اندازه فایل (کیلو بیت)	سطح	طول کلید (بیت)	دور	زمان رمزگذاری (میلی ثانیه)	زمان رمزگشایی (میلی ثانیه)
۱۰	۴	۴۰	۶	۳۲۸/۱۸	۳۳۰/۹۶
۲۰	۴	۴۰	۶	۴۸۰/۳۷	۵۷۳/۶۲
۳۰	۴	۴۰	۶	۶۶۱/۴۸	۸۲۳/۰۴

۴-۲ نتایج آزمون با تغییر کلید

سطح -۴

جدول ۳- مقایسه با توجه به کلید

کلید	سطح	طول کلید (بیت)	دور	زمان رمزگذاری (میلی ثانیه)	زمان رمزگشایی (میلی ثانیه)
Helloworld	۴	۸۰	۹	۱۸۱/۱۲	۱۸۱/۱۹
a	۴	۸	۴	۱۴۴/۷۶	۱۴۵/۸۲
ab	۴	۱۶	۵	۱۵۹/۰۹	۱۵۹/۴۳

۴-۳ نتایج آزمون با تغییر سطح

پسورد – sliit

جدول ۴- مقایسه با توجه به سطح

سطح	طول کلید (بیت)	دور	زمان رمزگذاری (میلی ثانیه)	زمان رمزگشایی (میلی ثانیه)
۲	۴۰	۴	۱۶۱/۳۰	۱۶۲/۶۶
۴	۴۰	۶	۱۶۷/۸۴	۱۶۹/۹۲
۶	۴۰	۸	۲۴۷/۴۳	۳۰۳/۰۶

۴-۴ نتایج آزمون با تغییر کلید و سطح

جدول ۵- مقایسه با توجه به سطح و کلید

کلید	سطح	طول کلید (بیت)	دور	زمان رمزگذاری (میلی ثانیه)	زمان رمزگشایی (میلی ثانیه)
a	۲	۸	۲	۱۵۳/۹۷	۲۷۹/۵۹
ab	۶	۱۶	۷	۱۵۹/۶۸	۲۸۰/۳۰
abc	۴	۲۴	۴	۲۶۷/۹۹	۳۲۰/۷۵

۴-۵ مقایسه با DES

جدول ۶- مقایسه با DES

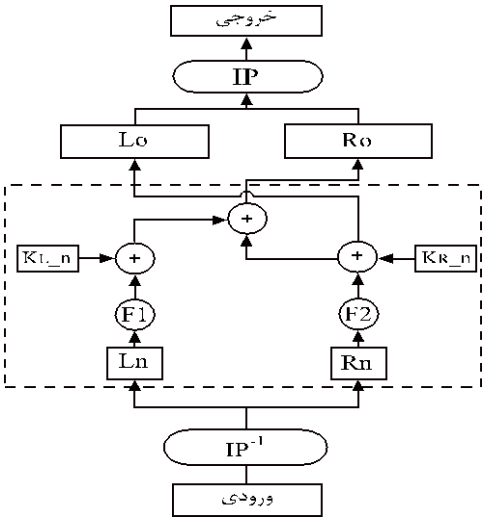
رمزگذاری DES (میلی ثانیه)	الگوریتم رمزگذاری ما (میلی ثانیه)	رمزگشایی DES (میلی ثانیه)	الگوریتم رمزگشایی ما (میلی ثانیه)
۵۶۱	۱۷۴	۱۱۰۱	۲۳۸
۵۶۲	۳۵۶	۱۸۱۷	۳۹۲
۵۶۵	۶۱۳	۲۲۸۷	۵۵۲

۵. نتیجه‌گیری

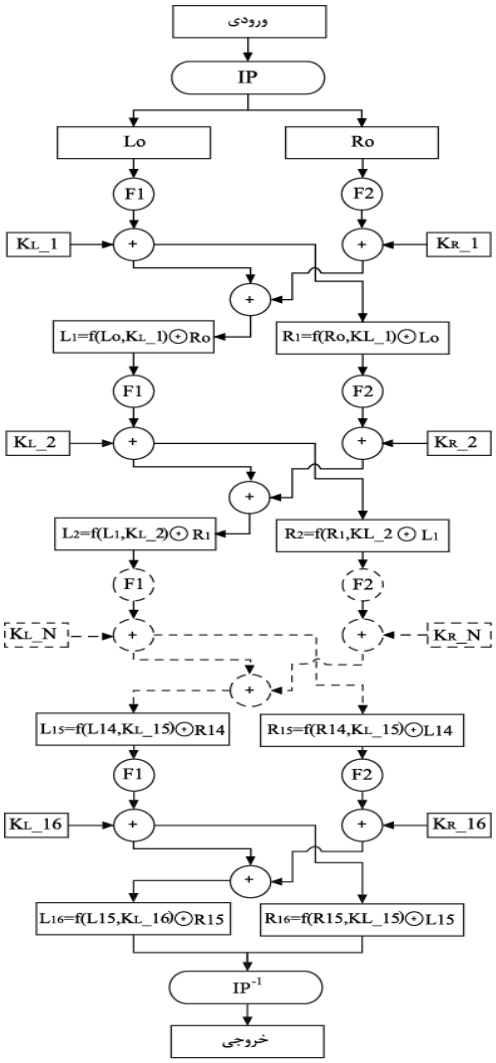
امنیت داده برای محدود کردن تهدیدات اطلاعاتی در سطوح کاربردی در یک سازمان، شرکت، مراکز حساس حکومتی و یا یک درگیری نظامی، حائز اهمیت فراوان می‌باشد. بنابراین ارائه الگوریتم‌های رمزنگاری قوی و پیشرفته از اهمیت زیادی برخوردار است. هدف از نوشتن این مقاله اجرای یک الگوریتم پیشرفته به کمک منطق فازی می‌باشد. ما در این مقاله الگوریتمی را معرفی کردیم که دارای سطح پردازش پایین و سطح امنیت بالا مطابق با نیاز کاربر بوده و همچنین از الگوریتم‌های رمزنگاری موجود، پیشرفته‌تر می‌باشد.

۶. مراجع

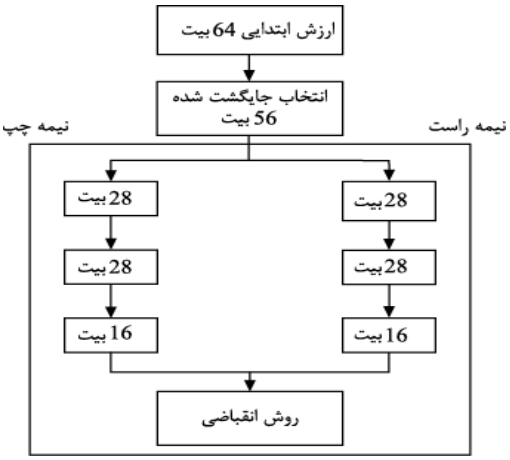
1. Schneier, B. (1994), "*Applied Cryptography,"* John Wiley & Sons, New York
2. Schneier, B. (1994), "*Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) Fast Software Encryption,"* Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag,
3. Zadeh, L.A. (1965), "*Fuzzy sets*, Information and Control," 338-353.
4. *TrueCrypt*, Thursday 15 July 2010, [online] http://www.truecrypt.org
5. *Microsoft Technet* , [online] http://technet.microsoft.com
6. Jones, D.W, "*Data Compression and Encryption Algorithms,"* http://www.cs.uiowa.edu
7. "Fuzzy Logic: An Introduction [online]," http://www.seattlerobotics.org
8. "Europe Gets into Fuzzy Logic," (1991), Electronics Engineering Times



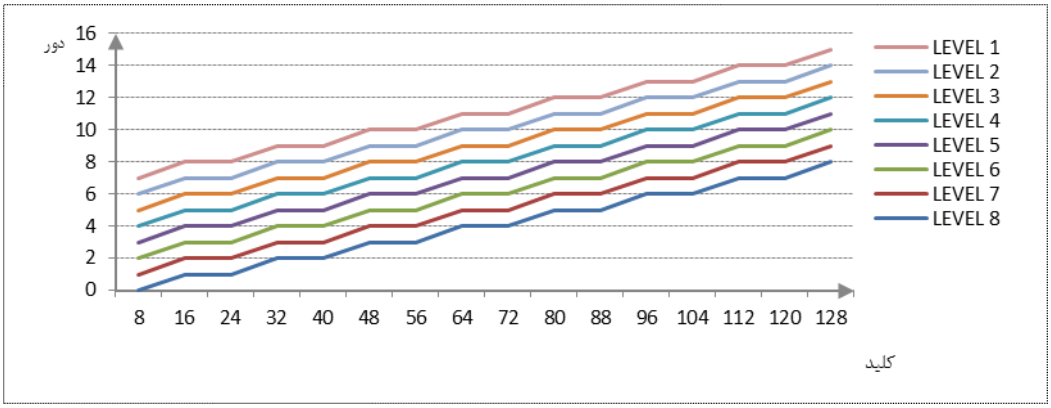
شکل ۱- الگوریتم رمزنگاری



شکل ۲- الگوریتم تولید کلید



شکل ۳- دیاگرام رمزگشایی



شکل ۴- دیاگرام منطق فازی