معرفی روش جدید رمزنگاری مبتنی بر تولید متن رمز شده متغیر

مهرداد زبیری[۱]، دانشجوی دکتری؛ بابک مظلوم نژاد میبدی[۲]، استادیار

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه شهید بهشتی - تهران - ایران - m_zobeiri@sbu.ac.ir
۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه شهید بهشتی - تهران - ایران - b-mazloom@sbu.ac.ir

چکیده: با گذشت زمان، کامپیوتر در امور بیشتری از زندگی انسان رخنه می‌کند، به همین سبب در عصر کامپیوتر نیاز به حفاظت از اطلاعات شخصی بیش از پیش احساس می شود. علم رمزنگاری برای حفاظت از این اطلاعات بکار می‌آید، طوری که اگر سیستمی تحت حمله قرار گرفت امکان دستر سی به اطلاعات غیرممکن با شد. در این مقاله یک روش جدید رمزنگاری برای متن معرفی می‌گردد که ایده آن، استفاده از روش‌های تئوری کدینگ در رمزنگاری می‌باشد. در این روش به کمک علم تئوری اعداد و محاسبات پیمانه‌ای، یک روش جهت تولید جعبه‌های جایگشت پویا و جعبه‌های جانشینی پویا بکار گرفته می شود. در ادامه به کمک علم تئوری اطلاعات و کدینگ، خطاهای تصادفی عمدی به متن اضافه می‌گردد. خطاهای عمدی ایجاد شده می‌بایست در زمان رمزگشایی به کمک روش‌های دی‌کدینگ کشف و تصحیح شود. تولید یک روش مستقل رمزنگاری با الگوی استفاده از خطای عمدی در کنار استفاده از کلید رمزکوتاه (۲۵۶ بیت)، مزیت این روش نسبت به روش‌های هم الگو می‌باشد. بعلاوه این روش قادر خواهد بود در هر بار اجرای الگوریتم رمزنگاری، برای یک متن آشکار ثابت و یک کلید رمزنگاری ثابت، متن رمز شده متفاوتی، با فاصله همینگ نزدیک به ۵۰ درصد، نسبت به اجرای قبل تولید نماید.

واژه‌های کلیدی: رمزنگاری بلوکی، محاسبات پیمانه‌ای، علم تئوری اطلاعات و کدینگ، جعبه‌جایگشت پویا، جعبه‌جانشینی پویا، اعداد اول، کلید رمزمتقارن.

Introducing New Cryptography Method to Make Variable Ciphertext

M. Zobeiri[1], PhD Student; B. Mazloom-Nezhad Meybodi[2], Assistant Professor

1- Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran, Email: m_zobeiri@sbu.ac.ir
2- Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran, Email: b-mazloom@sbu.ac.ir

Abstract: Through the passage of time, computer technology has been considered as a pivotal element in human life. Because of this point, it is more evident that information security gained a prominent position. Cryptography is utilized to secure information in a way that information became impenetrable under attack. In this paper, a new method of encryption is introduced according to the theoretical channel coding method and modular calculations. In this method, with the help of science of numerical theory and modular calculations, a method is used to generate dynamic permutation boxes and dynamic subsituation boxes. Then, with the help of information theory and coding, random errors are added to the text. Generated random errors should be detected and corrected at decryption time using decoding methods. The production of an independent cryptographic method with the pattern of intentional error use, along with the use of a shortcut key (256 bits), has the advantage of this method over the pattern-matching methods.This method can perform an encryption algorithm in which the performance of algorithm for a fixed text and a fixed encryption key, produce a differentiated encrypted text, with Hamming distance nearly 50%, in comparison to the previous performances.

Keywords: Block encryption, Modular calculations, Information theory and coding, Dynamic P-Box, Dynamic S-Box, Prime numbers, Symmetric key.

تاریخ ارسال مقاله: ۱۳۹۶/۷/۳۰
تاریخ اصلاح مقاله: ۱۳۹۶/۱۱/۲۹، ۱۳۹۷/۲/۰۹، ۱۳۹۷/۶/۱۸
تاریخ پذیرش مقاله:۱۳۹۷/۶/۲۳
نام نویسنده مسئول: بابک مظلوم نژاد میبدی
نشانی نویسنده مسئول: ایران - تهران - ولنجک – میدان دانشجو - دانشگاه شهید بهشتی - دانشکده مهندسی برق و کامپیوتر.

۱- مقدمه

الگوریتم رمزنگاری موفق، الگوریتمی است که در مقابل حملات جهت شکستن آن مقاوم باشد. الگوریتم پیشنهادی این مقاله در دسته الگوریتم‌های رمزنگاری متقارن قرار می‌گیرد. هر سیستم رمزنگاری متقارن، در بالاترین سطح ممکن، باید دارای دو ویژگی «پخش و پراکنده‌سازی» و «گمراه‌کنندگی» باشد. این دو ویژگی در سال ۱۹۴۹ توسط کلود شانون [۱] معرفی و از آن پس مورد استفاده قرار گرفت. از دیدگاه تئوری، سیستم رمزنگار باید شاخص‌های آماری متن آشکار را بر روی کل متن رمز شده، توزیع و پراکنده کند. به‌بیان‌دیگر یک سیستم رمزنگاری، هرگز نباید ویژگی‌های آماری متن را به هر نحو در خروجی رمز شده منتقل کند. بدین ترتیب هرگاه خروجی یک سیستم رمزنگار تحلیل آماری شود، نباید هیچ‌گونه همبستگی بین بیت‌های خروجی، کلید و بیت‌های متن مشاهده گردد.

در روش پیشنهادی، برای متن و کلید رمزنگاری ثابت، با هر بار اجرای الگوریتم رمزنگاری، متن رمز شده متفاوتی نسبت به اجرای قبل تولید می‌شود و این تفاوت کاملاً تصادفی است و از الگوریتم خاصی تبعیت نمی‌کند . این ویژگی باعث مقاوم شدن روش معرفی شده در مقابل حملات رایج رمزنگاری می‌شود. برای ایجاد چنین قابلیتی، در الگوریتم پیشنهادی، از علم تئوری اطلاعات و کدینگ کمک گرفته شده است[۲] که در بخش مربوطه توضیحات کامل ارائه خواهد شد.

ترکیب رمزنگاری و تئوری کدینگ، در سامانه‌های توأم رمزنگاری و کدگذاری کانال دارای سابقه می‌باشد. در زمان ارسال داده‌ها در کانال، ازیک‌طرف برای امنیت داده‌ها می‌بایست عمل رمزنگاری انجام شود و از طرف دیگر برای کشف خطاهای کانال در گیرنده باید از روش‌های تئوری کدینگ استفاده شود. به‌کارگیری الگوریتم‌های رمزنگاری و کدگذاری کانال به‌صورت مجزا در سامانه‌های مخابراتی از جمله تلفن‌های همراه، ماهواره‌های مخابراتی می‌تواند بالقوه باعث کاهش کارایی (افزایش پیچیدگی، افت سرعت و کاهش بازدهی مصرف انرژی) شود. یکی از راهکارهای تأمین امنیت و افزایش کارایی سامانه‌های مخابراتی، اجرای توأم الگوریتم های رمزنگاری و کدگذاری کانال به‌صورت یک فرآیند واحد است، به‌نحوی‌که امنیت و کارایی به‌طور هم‌زمان تا سرحد امکان برآورده شوند.

در این روش‌ها، برای استفاده از این سیستم ترکیبی، از کدهای کدینگ با طول چند صد کیلوبیت تا چند مگابیت استفاده می‌گردد. دلیل این امر آن است که خطاهای عمدی اضافه شده برای ایجاد امنیت نباید کم باشد و از طرفی با روش‌های کدینگ، کشف و اصلاح خطاهایی در حدود ۱۰ در صد طول کد ارسالی امکان‌پذیر است. همین موضوع باعث استفاده از کد با طول‌های بزرگ می‌گردد. به دلیل مشابه در این الگوریتم‌ها به علت استفاده از کدهای بزرگ، به‌ناچار باید طول بلوک انتخابی جهت ارسال و رمزنگاری نیز بزرگ باشد. از طرف دیگر در این‌گونه روش‌ها، برای افزایش امنیت، کدکدینگ انتخابی به همراه اطلاعات لازم دیگر در کلید رمزنگاری قرار می‌گیرد. این موضوع سبب افزایش طول کلید رمزنگاری در حد طول کد انتخابی تا چند مگابیت می‌شود.

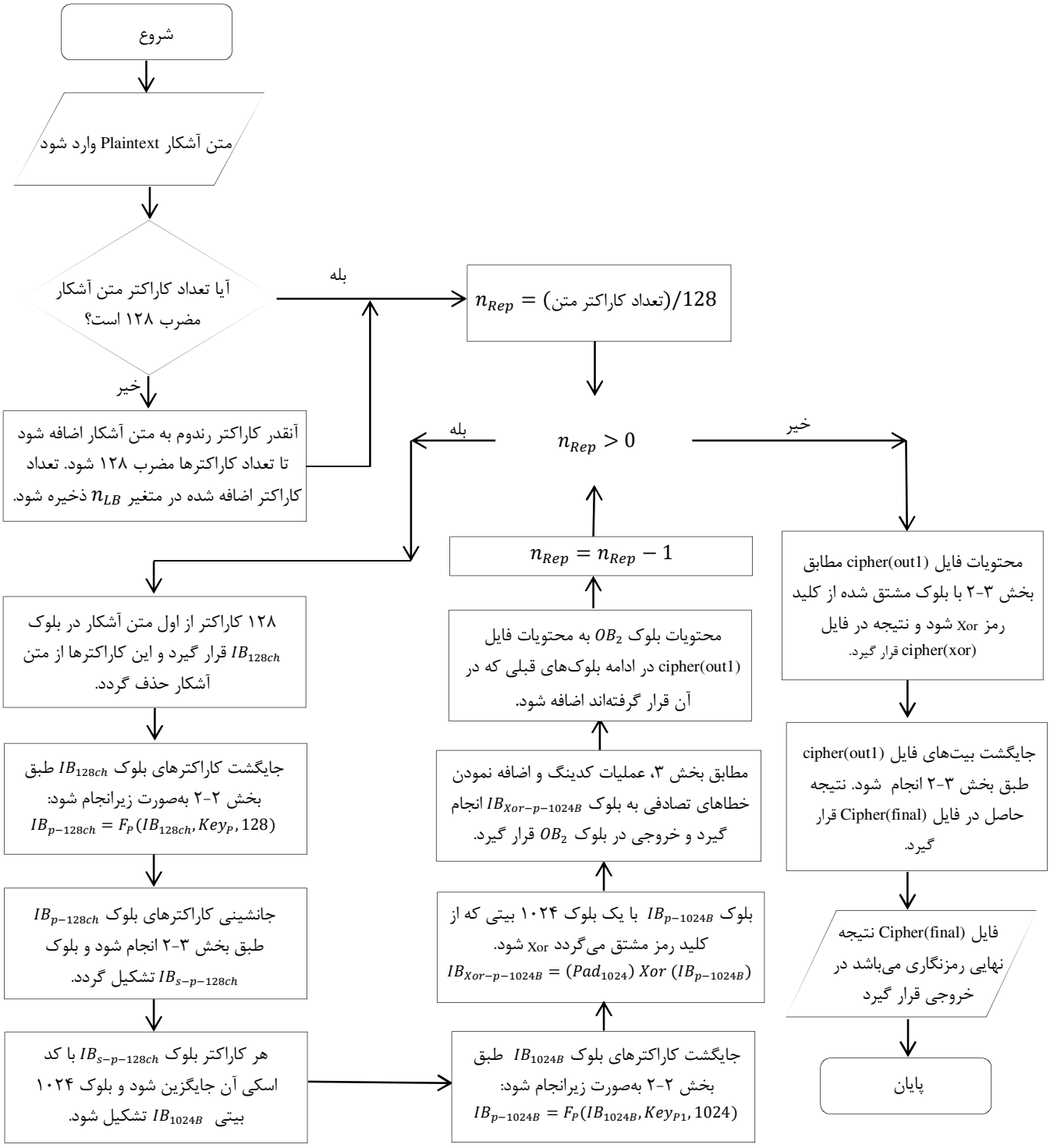
در مقاله حاضر، در کنار استفاده از جعبه‌های جایگشت و جانشینی پویا، از تئوری کدینگ برای ایجاد متن رمزشده متفاوت نسبت به اجرای قبل استفاده شد. در این روش، با ارائه راهکاری جدید، از کدهای کدینگ با طول چند ده بیت تا چند صد بیت جهت عملیات کدینگ استفاده می شود. در کنار کاهش طول کدهای کدینگ استفاده شده، نتایج نشان از ایجاد فاصله همینگ نزدیک به ۵۰ در صد در هر اجرای عملیات رمزنگاری نسبت به اجراهای قبل با یک کلید ثابت دارد. همچنین با کاهش طول کدهای کدینگ، طول بلوک انتخابی نیز کاهش یافت و جهت عملیات رمزنگاری از بلوک‌های۱۲۸ کاراکتری استفاده شد. این طول بلوک ورودی نسبت به بلوک‌های چند صدکیلوبیتی تا چند مگابیتی روش‌های ترکیبی بسیار کارآمد می‌باشد.

در این روش، ابتدا یک متن با طول دلخواه شامل کاراکترهای متنی، به‌عنوان متن آشکار در اختیار الگوریتم قرار می‌گیرد. ازآنجایی‌که این روش به‌صورت بلوکی عمل می‌نماید، در هر مرحله، تعدادی از کاراکترهای متن که هنوز رمز نشده‌اند، توسط الگوریتم انتخاب می شوند و طی فرایندی به کمک کلید رمز به اعداد صفر و یک تبدیل می‌شوند. این فرایند که در بخش دوم توضیح داده می‌شود از الگوریتم جایگشت و جانشینی پویا و خاصی استفاده می‌شود تا ویژگی‌های آماری متن، به‌صورت توزیع شده و نرمال در بیاید.

در بخش سوم، عملیات کدینگ و تغییر عمدی بیت‌های متن به صورت گسترده طبق الگوریتم پیشنهادی انجام می شود. در حقیقت این بخش از الگوریتم قابلیت متفاوت بودن متن رمز شده در هر اجرا، نسبت به اجراهای قبل را ایجاد می‌کند. در انتهای این بخش از الگوریتم جایگشت بخش دو، برای افزایش پیچیدگی، مجدد استفاده می‌گردد. در پایان بخش سه، متن آشکار به متنی رمز شده بر اساس الگوریتم پیشنهادی تبدیل می‌شود. در بخش چهارم، نحوه تشکیل کلید متقارن رمزنگاری معرفی می‌گردد. در بخش پنجم، عملکرد الگوریتم پیشنهادی و در بخش ششم، نتایج حاصل از استفاده از الگوریتم بررسی خواهد شد. در شکل ۱ فلوچارت کلی اجرای این الگوریتم قابل مشاهده است.

۲- بلوک‌بندی کاراکترها و تبدیل هر بلوک کاراکتری به بلوک صفر و یک با ویژگی‌های آماری توزیع شده

یکی از روش‌های متداول حمله به متن‌های رمز شده، استفاده از تحلیل آماری حروف و ترکیب‌های آن‌ها در متن است[۳]. توزیع یکنواخت کاراکترها و ترکیب آن‌ها در متن ورودی، برای مقاومت متن در مقابل حمله بسیار مهم و تأثیرگذار است. به همین دلیل درسیستم‌های رمزنگاری مدرن، داده‌ها در چندین مرحله درهم سازی می‌شوند. در این مراحل علاوه بر استفاده از عملکرد پر قدرت Xor ، از اجزای درهم‌ساز متعدد دیگر نیز استفاده می شود. یکی از ساده‌ترین ابزارهای رمزنگاری که در ترکیب با دیگر اجزا، در روش‌های رمزنگاری مدرن و متقارن کاربرد دارد P-BOX می‌باشد.



شکل ۱: فلوچارت الگوریتم رمزنگاری

در یک عبارت کوتاه، P-BOX ابزاری است که ترتیب بیت ها و کاراکتر های ورودی را به هم می‌ریزد و آن ها را در خروجی ظاهر می‌کند. از ابزارهای رمزنگاری پرکاربرد دیگر S-BOX است. S-BOX هر عدد n بیتی را، به‌صورت یک‌به‌یک، به عددی n بیتی دیگر می‌نگارد. اینجانشینی بر اساس جدول نگاشت مورد نظر طراح انجام می‌گیرد.

در بسیاری از روش‌های رمزنگاری متقارن همچون DES و AES از جدول‌های ثابت برای جایگشت و جانشینی بهره می‌برند. علت آن است که اگر این جدول‌ها ثابت نباشند چگونگی جایگشت و جانشینی باید در کلید رمز تعیین گردد. این کار باعث افزایش نامناسب طول کلید رمزنگاری می‌شود.

در ادامه این بخش، نحوه استفاده از جعبه‌های جایگشت و جانشینی پویای مبتنی برکلید، برای این روش مطرح می‌شود.

۲-۱- بلوک‌بندی فایل متن آشکار

روش پیشنهادی در دسته روش‌های رمزنگاری بلوکی قرار می‌گیرد. فایل ورودی جهت رمزنگاری، فایلی متنی است. در ابتدای هر مرحله از الگوریتم، ۱۲۸ کاراکتر اول، از بین کاراکترهایی که هنوز جهت رمزنگاری انتخاب‌نشده‌اند، انتخاب می‌گردد و به‌عنوان یک بلوک ورودی در اختیار الگوریتم قرار می‌گیرد. همان‌طور که در بخش‌های بعدی توضیح داده می‌شود، این الگوریتم می‌تواند نسخه‌های متمایزی داشته باشد. بسته به آنکه کدام نسخه اجرا می‌شود، طول بلوک ورودی از نظر تعداد کاراکتر می‌تواند مقادیر دیگری نیز بپذیرد.

درصورتی‌که تعداد کاراکترهای کل متن آشکار مضربی از ۱۲۸ نباشد، تعدادی کاراکتر تصادفی به انتهای متن آشکار اضافه می‌شود تا طول بلوک آخر نیز به ۱۲۸ کاراکتر تبدیل گردد. $n_{LB(Last\ Block)}$ برابر تعداد کاراکترهای اضافه شده به بلوک آخر متن آشکار می‌باشد. بدیهی است که $0 \le n_{LB} \le 127$ است، مقدار n_{LB} در کلید رمز قرار خواهد گرفت تا در زمان رمزگشایی، متن آشکار به‌درستی محاسبه شود.

۲-۲- جایگشت کاراکترهای بلوک ورودی

در روش رمزنگاری پیشنهادی از جدول جایگشت و جانشینی ثابتی استفاده نمی‌گردد. همین امر سبب افزایش میزان پیچیدگی الگوریتم حاضر و توانمند شدن آن در مقابله با حمله‌ها می‌باشد. به عبارتی جایگشت کاراکترها در این مرحله و بیت‌ها در مراحل بعد تابعی از کلید رمز خواهد بود. برای عدم افزایش نامطلوب کلید رمز، از روش تولید P-BOX براساس محاسبات پیمانه‌ای، بر اساس مقاله تولید جدول جانشینی و جایگشت پویا [۴]، در این الگوریتم استفاده شد.

در مقاله ذکر شده، طول بلوک ورودی که قرار است جای خانه‌های آن تعویض گردد، n_1 می‌باشد. در این روش، جابجایی خانه‌های ورودی برحسب ۴ عدد ثابت A',B',C',D' صورت می‌گیرد که آن‌ها به‌عنوان قسمتی از رمز در کلید رمز قرار خواهند گرفت. خانه‌های بلوک ورودی از ۱ تا n_1 شماره گذاری می‌شوند. یک بلوک خروجی با n_1 خانه در نظر گرفته می‌شود. در هر مرحله، شماره یکی از خانه‌های بلوک ورودی که هنوز انتخاب نشده است، براساس کلید انتخاب می‌گردد و محتویات آن در اولین خانه خالی بلوک خروجی قرار می‌گیرد. این عمل n_1 بار تکرار می‌گردد تا عمل جایگشت تمام n_1 خانه بلوک ورودی، انجام شود. این عمل جایگشت با تابع $F_P(Input\ Block, Key_P, n_1)$ نشان داده می شود که Key_P همان ۴ عدد ثابت A',B',C',D' می‌باشند و n_1 طول بلوک ورودی جهت جابجایی خانه‌های آن است. همان‌طورکه در بخش ۲-۱ مطرح شد، ۱۲۸ کاراکتر از متن آشکار انتخاب می‌شود، بنابراین $n_1 = 128$ می‌باشد. این کاراکترها به ترتیب در خانه‌های بلوک ورودی به نام IB_{128ch} قرار می‌گیرد.

برای استفاده از الگوریتم P-Box اشاره شده در این بخش، نیاز به چهار ورودی A'، B'، C' و D' جهت جابجایی خانه‌های بلوک ورودی می‌باشد. هر کدام از این اعداد به دلخواه از صفر تا ۱۰۲۳ انتخاب می شوند. Key_P از کنار هم قرار گرفتن این چهار متغیر ۱۰ بیتی ایجاد می‌شود. Key_P دارای ۴۰ بیت دلخواه می‌باشد و مقدار آن در کلید رمزنگاری قرار می‌گیرد. چگونگی پیاده‌سازی این الگوریتم در پیوست یک ارایه شده است.

جایگشت کاراکترهای داخل بلوک ورودی بر اساس تابع F_P صورت می‌گیرد و بلوک حاصل از جایگشت کاراکترهای ورودی $IB_{p-128ch}$ نامیده می‌شود. این بلوک شامل ۱۲۸ کاراکتر ورودی که برحسب کلید جایگشت داده شده‌اند می‌باشد.

لازم به تذکر است اعداد A'، B'، C' و D' به عبارتی Key_P که در کلید رمز قرار می‌گیرند، برای تمام بلوک‌های ۱۲۸ کاراکتری که قرار است رمز شوند یکسان است. به عبارتی جدول جایگشت ثابتی جهت تمام بلوک‌ها به‌وجود می‌آید. پیش از اجرای الگوریتم، به کمک کلید رمز، یک‌بار این جدول جایگشت ۱۲۸ کاراکتری محاسبه می‌گردد. بنابراین محاسبات جدول جایگشت که بار محاسباتی بالایی دارد، تنها یک‌بار انجام می‌شود و سر باری برای الگوریتم رمزنگاری پیشنهادی به‌حساب نمی‌آید.

$$IB_{p-128ch} = F_P(IB_{128ch}, Key_P, 128) \qquad (۱)$$

۲-۳- جانشینی کاراکترهای بلوک ورودی از مرحله قبل

با جایگشت کاراکترها که در بخش قبل طبق کلید رمز انجام شد، ترکیب و محل قرارگیری تمام کاراکترها به‌طور کامل به‌هم می‌ریزد. بنابراین هیچ وابستگی ۲ یا ۳ حرفی بین متن حاضر وجود ندارد، اما همچنان می‌تواند مورد تحلیل آماری تک‌حرفی قرار بگیرد[۵]. بر همین اساس در این بخش، هر کاراکتر متن برحسب مکان قرارگیری، به کاراکتری دیگر تبدیل می‌گردد[۶].

$IB_{p-128ch}$ بلوک ورودی این مرحله است. برای هر خانه این بلوک، یک عدد بین صفر تا ۱۲۷ انتخاب می‌شود. این عدد به کد اسکی کاراکتری که در آن خانه قرار دارد اضافه می‌شود و آن کاراکتر را به کاراکتر دیگری تبدیل می‌کند. به‌عنوان نمونه اگر در خانه اول کاراکتر I باشد و عددی که به کد اسکی کاراکتر این خانه اضافه می‌شود۳ باشد، این کاراکتر به کاراکتر L تبدیل می‌گردد.

برای افزایش پیچیدگی این عملیات، عمل جانشینی کاراکترها برحسب کلید رمز انجام می شود و طبق الگوریتم زیر که تنها ۱۰ بیت کلید رمز را به خود اختصاص می‌دهد، انجام می‌گیرد.

عدد ۱۰ بیتی E' به دلخواه بین صفر تا ۱۰۲۳ انتخاب می شود که این عدد نیز در کلید رمز قرار می‌گیرد. عدد E برابر $E' + 31$ قرار می‌گیرد. این عمل تضمین می‌کند حداقل مقدار E برابر ۳۱ می‌باشد. در هر مرحله جایگزینی کاراکترهای بلوک ورودی برحسب مکان

قرارگیری آن انجام می‌شـــود. در مرحله k ام ، طبق معادله ۲ میزان تغییر کاراکتر خانه k ام تعیین می‌گردد:

$$\text{mod } 127(E-K)^{(E+K)} \qquad (۲)$$

مقدار تغییر کاراکتر در مرحله k ام عددی بین صفر تا ۱۲۶ خواهد بود و این مقدار به کد اســکی کاراکتر واقع در خانه k ام بلوک $IB_{p-128ch}$ اضافه می‌شود. اگر حاصل عددی کوچک‌تر یا مساوی ۱۲۷ شد که کاراکتر مربوطه مشخص می‌گردد، در غیر این صورت باقیمانده عدد حاصـل بر ۱۲۸ به‌دسـت می‌آید که آن عدد، کاراکتر مورد نظر را مشخص می‌کند. علت محاسبه باقیمانده بر عدد ۱۲۷ در معادله ۲ آن اســت که عدد ۱۲۷ عددی اول اســت. بنابراین در محاســبه مقدار باقیمانده، از خواص محاسبات پیمانه‌ای می‌توان کمک گرفت، به همین جهت حجم عملیات به‌شدت کاهش می‌یابد.

لازم به ذکر اســت که در این بخش، تعداد تغییر هر خانه بلوک $IB_{p-128ch}$ برای تمام بلوک‌های بعدی که قرار است رمز شود یکسان اســت. بنابراین تنها یک‌بار این محاســبه انجام خواهد شــد. به عبارتی انجام این محاسبات، سرباری برای الگوریتم پیشنهاد نمی‌باشد. بلوک حاصـــل از جایگزینی کاراکترهای جدید، همچنان یک بلوک ۱۲۸ کاراکتری می‌باشد و نام این بلوک $IB_{s-p-128ch}$ انتخاب می‌گردد.

۴-۲- جایگزینی هر کاراکتر با کد اسکی و جایگشت بیت‌های تولید شده

در ادامه روش پیشـــنهادی، نیاز به تبدیل کاراکترها به اعداد دو دویی (مبنای ۲) می‌باشد. از کد اسکی جهت این تبدیل استفاده می‌شود. به عبارتی معادل هر کاراکتر که در خانه $IB_{s-p-128ch}$ قرار دارد، یک کد ۸ بیتی جایگزین می‌شـــود. بلوک جدید با نام IB_{1024B} شـــامل ۱۰۲۴=۸×۱۲۸ بیت خواهد بود. برای افزایش پیچیدگی الگوریتم بار دیگر از الگوریتم P-BOX معرفی شده در بخش ۲-۲ استفاده می‌شود:

$$IB_{p-1024B} = F_P(IB_{1024B}, Key_{P1}, 1024) \qquad (۳)$$

که در آن Key_{P1} نیز یک کلید ۴۰ بیتی کاملاً دلخواه، شامل ۴ عدد ۱۰ بیتی A_1'، B_1'، C_1' و D_1' می‌باشـــد. این اعداد پس از انتخاب در کلید رمز قرار می‌گیرند.

۵-۲- انجام عمل Xor بیت‌های بلوک ورودی با دنباله عددی حاصل از کلید

در ســـیســـتم‌های رمزنگاری تابع Xor ابزاری برای افزایش دو ویژگی «پخش و پراکنده سازی» و «گمراه‌کنندگی» می‌باشـد. کلود شـانون اثبات نمود پس از Xor کلید با متن اصـلی پیام، درصـورتی‌که انتخاب کلید کاملاً تصادفی باشد، هیچ اطلاعات یا شاخص آماری از متن رمز شده مشاهده نخواهد شد.

در ادامه الگوریتم رمزنگاری، بلوک $IB_{p-1024B}$ با یک بلوک ۱۰۲۴ بیتی که از کلید رمز مشـتق می‌گردد Xor می‌شـود. این عمل مقاومت الگوریتم را در مقابل حملات افزایش می‌دهد. بلوک ۱۰۲۴ بیتی به نام Pad_{1024} جهت عملیات Xor از کلید رمز محاسبه می‌شود:

$$IB_{Xor-p-1024B} = (Pad_{1024})\ Xor\ (IB_{p-1024B}) \qquad (۴)$$

برای ایجاد بلوک Pad_{1024} به‌صورت زیر اقدام می‌گردد:

۱) عدد دلخواه F به طول ۳۲ بیت انتخاب می‌گردد. این عدد در کلید رمز قرار خواهد گرفت.

۲) عدد G به طول ۶ بیت به‌صـــورت دلخواه انتخاب می‌گردد. این عدد نیز در کلید رمز قرار خواهد گرفت.

۳) بلوک ۱۰۲۴ بیتی Pad به‌صورت زیر تعریف می‌گردد:

بلوک pad به ۳۲ بلوک ۳۲ بیتی به نام Pad(0)، Pad(1)، ... و Pad(31) مطابق جدول یک تقسیم می‌گردد.

جدول ۱: تقسیم بلوک Pad به ۳۲ بلوک ۳۲ بیتی

Pad=	Pad(0)	Pad(1)	…	Pad(31)
شماره بیت‌ها در Pad	۳۲-۱	٦٤-۳۳	…	۱۰۲٤-۹۹۲

در ابتدا Pad(0) برابر عدد ۳۲ بیتی F قرار می‌گیرد، ســـپس هر Pad(i) برابر i بار چرخش به راست عدد ۳۲ بیتی F می‌شود.

۴) عدد m به‌صـــورت m=G+0.9 تعریف می‌گردد. عدد m یک عدد گویای غیرمربع کامل می‌باشد. عدد m' برابر $\sqrt{m}$ قرار می‌گیرد. عدد m' یک عدد گنگ می‌باشـد (اثبات آن در پیوست دو مقاله قابل مشـاهده است.)، بنابراین تعداد اعشارهای آن بی‌شمار است. قسمت صحیح عدد m' را حذف کرده و عدد به‌جا مانده m'' نامیده می‌شود.

$$m'' = m' - [m'] \qquad (۵)$$

معادل باینری عدد m'' در مبنای ۲ محاسبه می‌گردد. از آنجا که تعداد اعشار عدد m'' بی‌شمار است، تعداد اعشار m'' در مبنای ۲ نیز بی‌شمار خواهد بود. ۱۰۲۴ رقم اول بعد از اعشـار عدد m'' در مبنای ۲ به‌عنوان متغیر Pad' انتخاب می‌گردد.

$$m'' = (0.a_1a_2a_3\cdots a_{1024}a_{1025}\cdots)_2$$
$$Pad' = a_1a_2a_3\cdots a_{1024} \qquad (۶)$$

۵) بلوک Pad_{1024} به‌صورت $Pad_{1024} = (Pad)Xor\ (Pad')$ به‌دست می‌آید.

لازم به ذکر است که Pad_{1024} در این الگوریتم تنها یک‌بار به‌دست می‌آید و در رمز کردن بلوک‌های دیگر ثابت باقی می‌ماند. بنابراین محاسـبات انجام شـده سـربار الگوریتم نمی‌باشـد. همچنین Pad_{1024} کاملاً به کلید رمز وابسته است و تغییر کلید رمز مربوط به آن، تغییرات اساسی در آن ایجاد می‌کند.

در انتهای این بخش بلوک $IB_{Xor-p-1024B}$ به دســـت می‌آید. این بلوک شامل ۱۰۲۴ بیت می‌باشد که تابعی پیچیده و کاملاً غیرخطی از ۱۲۸ کاراکتر متن آشکار و کلید رمز است. در شکل شماره دو الگوریتم اجرا شده تا اینجا قابل‌ملاحظه می‌باشد.

تعداد خانه‌های بلوک ایجاد شده	بلوک تولید شده بعد از اجرا مرحله	عنوان بخش	ورودی بخش و عملیات انجام شده
128	IB_{128ch}	بلوک‌بندی متن	متن آشکار
128	$IB_{p-128ch}$	جایگشت بلوک ورودی	$Key_P = f_1(Encription\ Key)$
128	$IB_{s-p-128ch}$	جانشینی کاراکترهای بلوک ورودی	$E = f_2(Encription\ Key)$
1024	IB_{1024B}	جایگزینی هر کاراکتر با مقدار کد اسکی آن	
1024	$IB_{p-1024B}$	جایگشت بیت‌های تولید شده	$Key_{P1} = f_3(Encription\ Key)$
1024	$IB_{Xor-p-1024B}$	Xor کردن با Pad	$(Pad_{1024})\ Xor\ (IB_{p-1024B})$

شکل ۲: الگوریتم اجرا شده در بخش دو

۳- استفاده از تئوری کدینگ و ایجاد خاصیت متغیر کردن متن رمز شده در هر بار اجرا

در کانال انتقال در مخابرات، در هنگام انتقال داده، ممکن است به علت وجود نویز در محیط، داده‌های ارسالی دچار تغییر شده و گیرنده آن‌ها را همراه خطا دریافت کند. به همین جهت، گیرنده باید به طریقی متوجه دریافت داده به همراه خطا شود. کنترل خطا در این حالت، به دو دسته کشف خطا و تصحیح خطا تقسیم می شود. در هر دو مورد، بیت‌هایی به داده‌های در حال ارسال اضافه می‌شود تا به کمک آن‌ها وجود خطا در داده ارسالی کشف و در صورت امکان تصحیح شود، این عمل را کدینگ می‌نامند. روش‌های متعددی برای کدگذاری کانال در علم کدینگ مطرح می‌شود. یکی از روش‌های متداول آن، روش بلوک خطی می‌باشد. در این مقاله از این روش استفاده شده است[۹-۷].

کدهای بلوک خطی به صورت یک کد C(n,k) نشان داده می شوند. بنابراین تعداد k بیت به کدکننده وارد می‌شود و بعد از عملیات کدگذاری و اضافه نمودن تعداد مشخصی بیت به آن ، اطلاعات کد شده n بیتی حاصل می‌شود. اگر حداقل فاصله همینگ روش کدگذاری d_{min} در نظر گرفته شود، در انتقال داده اگر حداکثر s بیت دچار خطا شوند $(s = d_{min} - 1)$، در گیرنده امکان کشف قطعی خطا وجود دارد. همچنین اگر تعداد بیت‌های خطا کوچک‌تر یا مساوی آستانه t باشد $(t = \left[\frac{d_{min}-1}{2}\right])$، گیرنده علاوه بر کشف خطا قادر است خطای رخ داده را به‌صورت قطعی تصحیح و کد اصلی ارسالی را کشف نماید. البته این موضوع به آن معنا نمی‌باشد که اگر تعداد خطاها بیش از t باشد، در گیرنده امکان تصحیح وجود ندارد، بلکه از نظر ریاضی به‌طورقطع نمی‌توان گفت کد اصلی حتماً کشف می‌شود.

در کدهای بلوک خطی، اگر پیام اولیه با بردار m نشان داده شود، ماتریس مولدی بنام **G** وجود دارد که با ضرب آن در پیام، کد متناظر آن تولید می‌شود. این کد را «کلمه کد» می‌نامند.

$$V = m \times \boldsymbol{G} \qquad (۷)$$

که در آن m شامل k بیت پیام و **G** ماتریس مولد و V کلمه کد می‌باشد.

در گیرنده فرض برآن است که $\mathbf{Z} = V + u_{error}$ دریافت می‌شود. u_{error} برداری معادل خطای ایجاد شده در کانال می‌با شد. الگوریتمی که بردار $\boldsymbol{V}$ را از روی بردار $\boldsymbol{Z}$ کشف می‌کند الگوریتم دی‌کدینگ نامیده می‌شود. الگوریتم‌های دی‌کدینگ متعددی وجود دارد.

با توجه به آنکه تنها از اعداد صفر و یک، به‌عنوان داده ورودی در بخش کدینگ مقاله حاضر استفاده می‌شود، در روش پیشنهادی از الگوریتم دی کدینگ Bit Flipping(BF) استفاده شده است. درصورتی‌که این الگوریتم بتواند خطاها را به‌درستی تشخیص دهد و بردار $\mathbf{Z_{new}}$ را به‌عنوان $\boldsymbol{V}$ کشف کند می‌بایست $\boldsymbol{Z_{new}} \times \boldsymbol{H}^T = 0$ شود. (ماتریس **H** را ماتریس دوگان **G** می‌نامند.)

ایجاد خطا و امکان تصحیح آن توسط روش دی‌کدینگ، مبنای روش پیشنهادی برای تولید متن رمز شده متغیر در این مقاله قرار گرفت. به کمک علم کدینگ، بیت‌های خطای عمدی و تصادفی به متن اضافه می‌شود. سپس تابعی در ادامه معرفی می‌گردد تا به کمک خطاهای تصادفی بالا تغییرات عمده‌ای در متن ایجاد کند. ازآنجایی‌که این خطاها تصادفی خواهد بود، در هر بار اجرا، متن رمز شده‌ای متمایز با اجراهای قبل به‌دست می‌آید. مسیر تحقق این هدف در ادامه تشریح می‌گردد.

۳-۱- کد کردن بلوک خروجی از بخش دو الگوریتم و اضافه نمودن بیت‌های خطای تصادفی

۱۲۸ کاراکتر انتخابی از متن آشکار که در انتهای بخش قبل به‌صورت بلوک $IB_{Xor-p-1024B}$ درآمده، در این بخش، بلوک IB_2 نامیده می‌شود. در مقاله حاضر از ۴ کد $C_0(126,76)$، $C_1(21,12)$، $C_2(16,11)$ و $C_3(74,46)$ جهت عملیات کدینگ استفاده می شود. با توجه به آنکه نسخه‌های متمایزی از این الگوریتم می‌تواند ارائه شود، می‌توان از کدهای دیگر با طولی متمایز و یا تعداد کد متمایز دیگر، مثلاً ۷ یا ۸ کد، استفاده شود.

در هر مرحله کد کردن، ابتدا یک کد طبق الگوریتمی که ارائه می شود برحسب کلید رمز انتخاب می‌گردد. با توجه به تعداد بیت‌های پیام آن کد، از بیت‌های کد نشده IB_2 انتخاب می‌گردد و کلمه کد مورد نظر تولید می‌شود. پس از اضافه نمودن خطاهای عمدی و گسترده به این کد، طبق الگوریتمی که ارائه خواهد شد، این کد در OB_2 در ادامه بیت‌های کد شده قبلی قرار می‌گیرد. در آخرین مرحله کد کردن بلوک IB_2 ، ممکن است تعداد بیت کافی در اختیار نباشد. به‌عنوان‌مثال فرض کنید C_3 انتخاب شود که نیاز به ۴۶ بیت دارد، اما

تنها ۳۰ بیت کد نشده از IB_2 باقیمانده باشد. در این صورت رقم صفر آن‌قدر به انتهای بیت‌های انتخابی مانده می شود تا بیت‌های لازم جهت استفاده از کد مورد نظر در دسترس باشد. این موضوع در زمان رمزگشایی خللی ایجاد نمی‌کند. در زمان رمزگشایی در آخرین مرحله می‌دانیم طول IB_2 برابر ۱۰۲۴ است و بیت‌های اضافه به‌دست‌آمده بیش از طول ۱۰۲۴ بیت، همان بیت‌های زائد اضافه شده می‌باشد.

در مرحله k ام کد کردن، برای انتخاب شماره کد انتخابی جهت عملیات کدینگ به‌صورت زیر اقدام می‌شود:

عدد دلخواه ۶ بیتی H' به‌تصادف انتخاب می‌گردد. عدد H' در کلید رمز قرار خواهد گرفت. عدد H برابر $H' + 11$ قرار می‌گیرد. فرض نمایید برای k امین دفعه قرار است تعدادی از بیت‌های IB_2 انتخاب و عمل کدگذاری انجام شود. شماره کد انتخابی در این مرحله به صورت زیر انتخاب می‌شود:

$$\left[(H-k)^{(H+k)} \bmod t_1\right] \bmod t_2 \qquad (۸)$$

که در آن k شماره مرحله و H عدد دلخواه انتخابی بعلاوه ۱۱ می‌باشد. مقدار t_2 برابر تعداد کدهای مورد استفاده است که در این مقاله $t_2 = 4$ می‌باشد. متغیر t_1 عددی اول بزرگ‌تر از t_2 انتخاب می‌شود. اول بودن عدد t_1 سبب می‌شود بتوان از محاسبات پیمانه‌ای کمک گرفت. در این مقاله t_1 برابر ۱۳ انتخاب‌شده است. در نسخه‌های متفاوت الگوریتم اجرایی، مقدار t_1 و t_2 ، برحسب نسخه، می‌تواند متفاوت انتخاب گردد.

لازم به ذکر است که ترتیب انتخاب کدها، جهت کدگذاری هر مرحله وابسته به کلید رمز می‌باشد و چون کلید رمز برای تمام بلوک‌ها یکسان است(عدد H یکسان است)، ترتیب انتخاب کدها جهت بلوک اول، در بلوک‌های بعدی نیز مورد استفاده قرار می‌گیرد. به همین سبب محاسبات انجام گرفته برای انتخاب ترتیب کدها، سرباری برای الگوریتم پیشنهادی نمی‌باشد.

مراحل زیر جهت کد کردن بلوک IB_2 و اضافه نمودن بیت‌های خطای تصادفی انجام می‌پذیرد:

1. عدد مناسبی به y نسبت داده می‌شود.

   اگر اولین مرحله از اولین بلوک برای رمزنگاری است، عدد ۶ بیتی I به دلخواه انتخاب می‌شود (عدد I در کلید رمز قرار می‌گیرد) و y برابر مقدار معادل آن در مبنای ۱۰ قرار می‌گیرد. در غیر اینصورت، از y خروجی بلوک قبل استفاده می‌شود.
2. متغیرهای k و x برابر صفر قرار می‌گیرد.
3. به عدد k یک واحد افزوده می‌شود.$(k = k + 1)$
4. مطابق معادله ۸ کد انتخابی مرحله k ام به دست می‌آید.
5. با توجه به ویژگی‌های $C_K(n_k, m_k)$ ، m_k بیت از بیت‌های انتخاب نشده IB_2 انتخاب می‌گردد و به کمک ماتریس مولد **G**، متناظر C_K، کلمه کد n_k بیتی V_k محاسبه می‌شود. $(\mathbf{V} = \boldsymbol{G} \times m)$
6. دو بردار خطا به نام u_e و u_{error} به طول n_k بیت که تمام بیت‌های آن‌ها صفر هستند تعریف می‌گردد.
7. به‌صورت تصادفی یکی از بیت‌های u_e که صفر می‌باشد انتخاب‌شده و مقدار آن برابر یک می‌شود. بردار **Z** به‌صورت $\mathbf{Z} = \mathbf{V} + u_e$ تعریف می‌شود..
8. بردار **Z** به الگوریتم دی‌کدینگ BF داده می‌شود تا بردار اصلی به دست آید.

   اگر عملیات دی کدینگ **V** را بازگرداند $u_{error} = u_e$ قرار می‌گیرد و به مرحله ۷ باز می‌گردیم، در غیر این صورت به مرحله ۹ می‌رویم.
9. بردار خطای u_{error} به بردار V افزوده می‌شود. $(\mathbf{V} = \mathbf{V} + u_{error})$

   بردار خطای u_{error} بردار خطایی است که می‌تواند به بردار **V** اضافه شود، به شرطی که طی عملیات دی‌کدینگ BF در الگوریتم رمزگشایی، حتماً بیت‌های خطای آن شناسایی و اصلاح شود. با توجه به کد انتخابی C_K، حداقل تعداد یک‌های این بردار خطا $\left[\frac{d_{min}-1}{2}\right]$ می‌باشد. مرحله ۷ آن‌قدر تکرار شده است که عملیات دی‌کدینگ بتواند خطاها را اصلاح کند.
10. بردار خطای $u_{extra-error}$ به طول n_k بیت که تمام بیت‌های آن صفر هستند تعریف می شود. مقدار جدید این خطا به صورت زیر محاسبه می‌گردد:

    i) C = y
    ii) C = C + 0.9
    iii) $d = \sqrt{c}\ ,\ d \in Q'$
    iv) عدد d عددی گنگ می‌باشد و دارای بی‌شمار رقم اعشار است. قسمت صحیح آن حذف می‌شود.$(d = d - [d])$

    عدد به‌جا مانده به مبنای ۲ تبدیل می‌شود. تعداد n_k تا از رقم‌های سمت راست ممیز عدد d در مبنای ۲، در $u_{extra-error}$ قرار می‌گیرد.

$$d = (0.d_1 d_2 d_3 \cdots d_{n_k})_2$$
$$u_{extra-error} = (d_1, d_2, d_3, \cdots, d_{n_k}),\ d_i = 0\ Or\ 1 \qquad (۹)$$

    لازم به ذکر است که بیت‌های d_1 تا d_{n_k} با توجه به مقدار تصادفی y محاسبه می‌شوند.
11. بردار خطای $u_{extra-error}$ به بردار $\boldsymbol{V}$ افزوده می‌شود.

$$V = V + u_{extra-error} \qquad (۱۰)$$

12. مقدار جدید y به‌صورت زیر برای مرحله بعدی محاسبه می‌گردد:

$$y = \sum_{i=1}^{n_k} u_{error}(i) \times i^2 \qquad (۱۱)$$

    که در آن $u_{error}(i)$ برابر بیت i ام بردار u_{error} می‌باشد.

    لازم به ذکر است در زمان رمزگشایی، در اولین مرحله مقدار y از کلید رمز به دست می‌آید. به کمک این مقدار، بردار $u_{extra-error}$ محاسبه می‌شود. با اضافه نمودن این خطا به بردار اول، خطای $u_{extra-error}$ که به پیام کد شده در زمان رمزنگاری اضافه شده است، جبران می شود. بعد از کشف خطاهای عمدی u_{error} توسط الگوریتم دی‌کدینگBF و تصحیح آن‌ها، بردار

اصـــلی به دســـت می‌آید. با مقایســـه بردار اصـــلی و بردار قبل از عملیات دی‌کدینگ BF، بردار خطای u_{error} حاصل می شود. به کمک این بردار مقدار y مرحله بعد به دست می‌آید.

13. مقدار x به‌اندازه m_k اضافه می‌شود.($x = x + m_k$)
همواره مقدار x نشان‌دهنده تعداد بیت‌های کد شده از بلوک IB_2 می‌باشد.

14. مقدار $\boldsymbol{V}$ (کد همراه خطای به‌دست‌آمده) در بلوک OB_2 ، در ادامه بیت‌های کد شده قبلی قرار می‌گیرد.

15. اگر $x \geq 1024$ باشـــد الگوریتم کدینگ پایان می‌پذیرد، در غیر اینصورت به مرحله ۳ برمی‌گردیم.

در شـــکل ســـه فلوچارت الگوریتم کد کردن بلوک IB_2 و اضـــافه نمودن بیت‌های خطای تصادفی قابل مشاهده است.

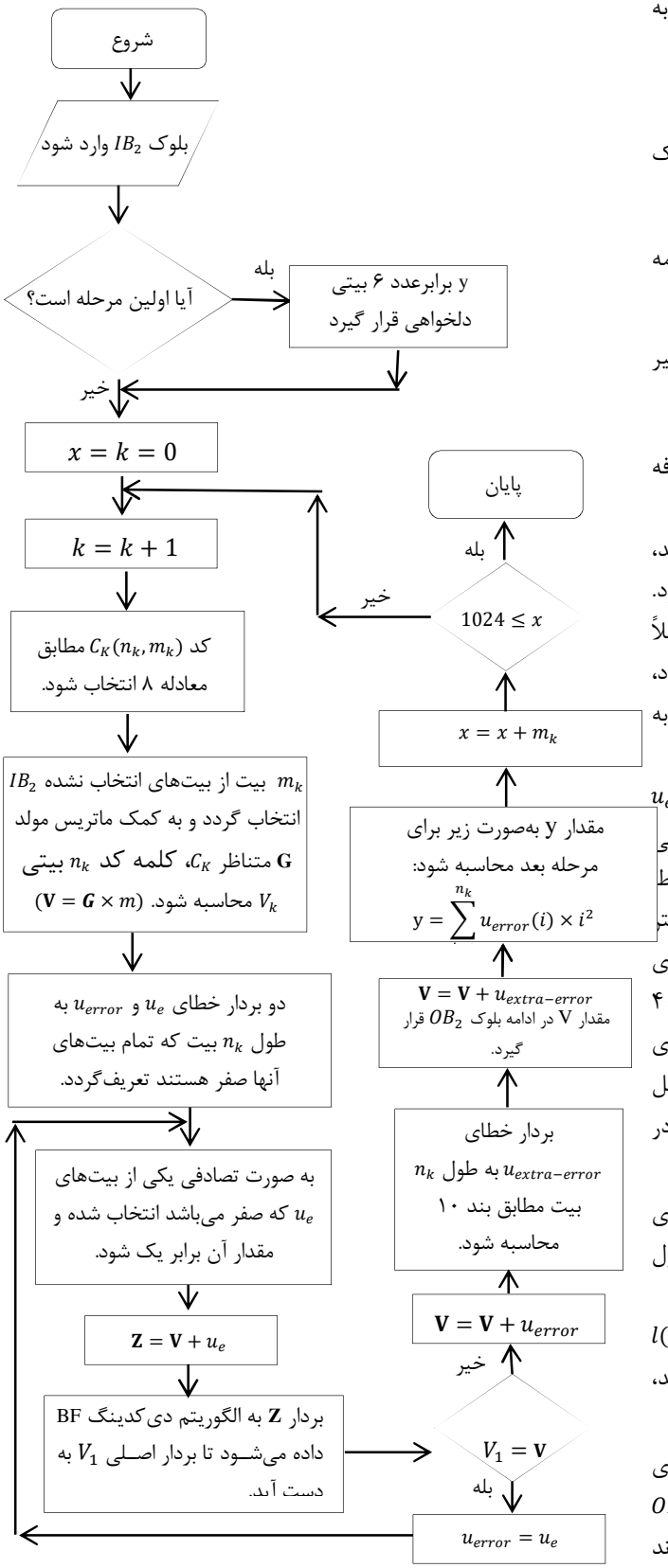
ازآنجایی که بیت های u_{error} کاملاً تصـــادفی انتخاب‌شـــده‌اند، به‌هیچ‌عنوان، مقدار y قابل محاســـبه توســـط رمزشـــکن نخواهد بود. بنابراین $u_{extra-error}$ در مرحله بعد، تابعی برحســـب مقادیر کاملاً تصادفی و غیرمحاسبه y این مرحله خواهد بود. این امر سبب می‌گردد، با هر بار اجرای الگوریتم، متن رمز شده کاملاً متمایزی از اجرای قبل به دست بیاید.

در اجرای الگوریتم بالا دو بردار خطای u_{error} و $u_{extra-error}$ معرفی گردید. همان‌طورکه گفته شـد، بیت‌های u_{error} کاملاً تصـادفی انتخاب می‌شوند. با توجه به d_{min} کد استفاده شده، وزن این بردار خطا (تعداد یک‌های بردار u_{error}) درصد چندانی از طول آن نمی‌باشد(کمتر از ۱۰ درصـــد). از آنجا که در این پروژه می‌بایســـت با هر بار اجرای الگوریتم، متن رمز شـــده جدید با اختلاف مناســـب بیتی (بیش از ۴۰ درصـــد) نســـبت به اجراهای دیگر حاصـــل شـــود، از بردار خطای $u_{extra-error}$ استفاده شد. مولفه‌های این بردار به مقدار y مرحله قبل وابســـته اســـت. به این ترتیب در هر اجرای جدید، تغییرات عمده‌ای در متن رمز شده نسبت به اجرای قبل ایجاد می‌شود.

در زمان کدگذاری با تغییر عدد دلخواه H' در کلید رمز، کدهای انتخابی در الگوریتم پیشنهادی متفاوت خواهد شد، به این ترتیب طول OB_2 که خود تابعی از کدهای انتخابی است، عددی ثابت نخواهد بود.

چنانچه طول OB_2 حاصـــل از رمزنگاری بلوک اول را ($l(OB_2)$ بنامیم، چون عدد H' عدد ثابتی برای تمام بلوک‌های رمز می‌باشـــد، طول تمام بلوک‌های کد شده نیز ($l(OB_2)$ خواهد بود.

فایل متنی بنام cipher(out1).txt ایجاد می‌شـــود. نتیجه رمزنگاری هر بلوک ۱۲۸ کاراکتری در بلوک OB_2 قرار می‌گیرد. این بلوک OB_2 در فایل cipher(out1) در ادامه بلوک‌های قبلی که در آن قرارگرفته‌اند قرار می‌گیرد. به عبارتی تا زمانی که رمز کل متن آشـــکار در فایل cipher(out1) قرار نگرفته، این کار تکرار می‌شود.



شکل ۳: فلوچارت الگوریتم کد کردن بلوک IB_2 و اضافه نمودن بیت‌های خطای تصادفی

۲-۳- انجام عملیات Xor و جایگشت نهایی جهت تولید متن رمزشده

آخرین قدم برای تکمیل فرایند رمزنگاری، آن اســت که بیت‌های فایل cipher(out1) ابتدا با بلوکی مشــتق شــده از کلید رمز xor شــوند و در آخرین مرحله بیت‌های حاصل جایگشت داده شوند.

ابتدا بیت های فایل cipher(out1) به بلوک های ۱۰۲۴ بیتی تقسیم می‌شوند. اگر تعداد بیت‌ها این فایل مضربی از ۱۰۲۴ نباشد، تعدادی بیت صــفر و یک به‌صــورت تصــادفی به انتهای فایل افزوده می‌گردد، تا مضــرب ۱۰۲۴ شــود. تعداد این بیت‌های اضــافه در متغیر $n_{added-bit}$ قرار می‌گیرد. بدیهی اســت که $0 \le n_{added-bit} \le 1023$ می‌باشد، مقدار $n_{added-bit}$ در کلید رمز قرار خواهد گرفت.

در ادامه عدد دلخواه ۹۶ بیتی J به‌صورت تصادفی انتخاب می‌گردد. این عدد در کلید رمز قرار خواهد گرفت. بلوک ۱۲۸ بیتی FX(final Xor) از کنار هم قرار دادن بیت‌های F و J به‌صــورت زیر محاســبه می‌گردد:

FX='J' +'F' (۱۲)

به‌طوری‌که عدد F همان عدد ۳۲ بیتی می‌باشد که در بخش ۲-۵ مورد ا ستفاده قرارگرفته ا ست. همان‌طورکه در آن بخش گفته شد این عدد نیز در کلید رمز قرار دارد.

فایل جدید متنی بنام cipher(xor) تعریف می‌گردد. از ابتدای فایل cipher(out1) تعداد ۱۲۸ بیت به ترتیب انتخاب‌شده و با محتوای بلوک FX، Xor می‌شــود و نتیجه آن در فایل cipher(xor) قرار می‌گیرد. این عمل تا انتهای فایل cipher(out1) انجام می‌شود.

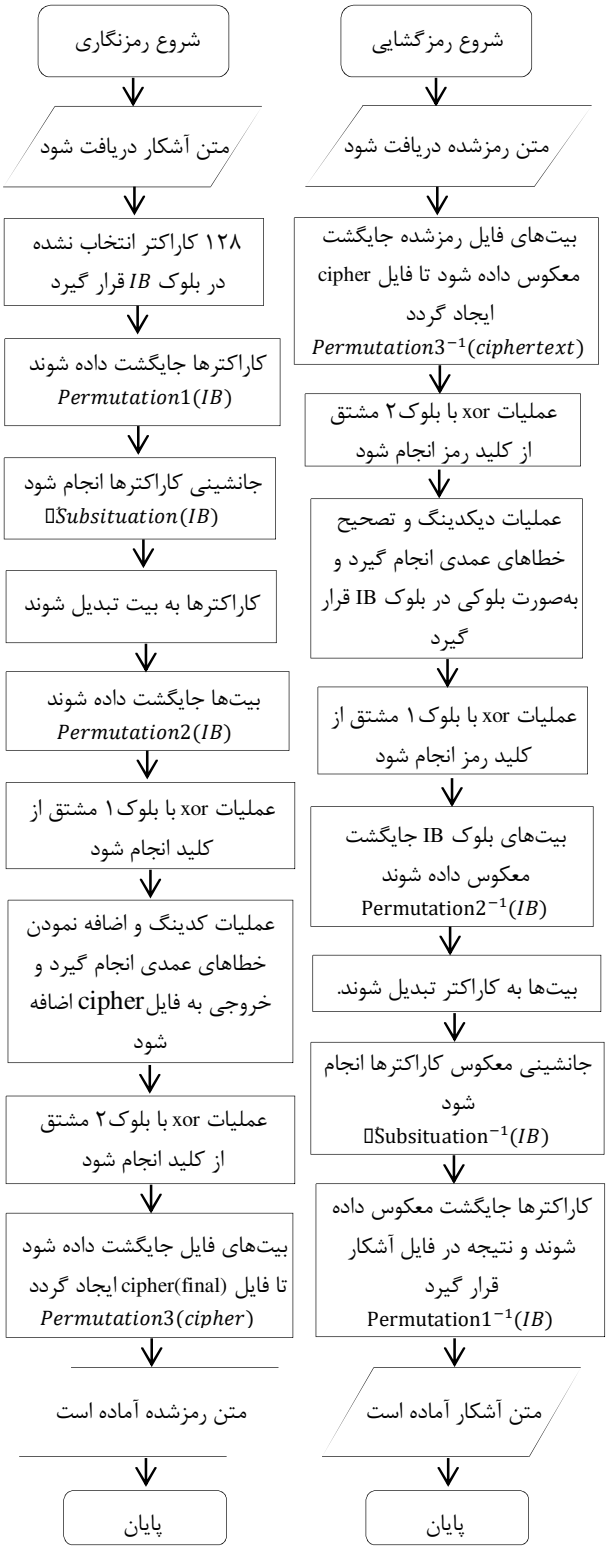
جایگشت نهایی:

در بخش ۲-۲، ۱۰۲۴ بیت بلوک ورودی IB_{1024B} طبق تابع F_P و کلید رمز جایگشت داده شد. بار دیگر از این تابع کمک گرفته می‌شود. فایل جدید متنی به نام Cipher(final) و بلوک ۱۰۲۴ بیتی دیگری به نام FP(final permutation) ایجاد می‌گردد. از ابتدای فایل cipher(xor)، تعداد ۱۰۲۴ بیت به ترتیب انتخاب‌شــده و در بلوک FP قرار می‌گیرد. نتیجه حاصــل از تابع $F_P(\mathrm{FP}, Key_{P1}, 1024)$ در فایل Cipher(final) قرار می‌گیرد. این عمل تا انتهای فایل cipher(xor) انجام می‌شــود. (Key_{P1} همان کلید بخش ۲-۴ است. بنابراین نیازی به محاسبه جدول جدید جایگشت وجود ندارد.)

فایل Cipher(final) شــامل متن رمز شــده یا به عبارتی متن نهایی الگوریتم پیشنهادی می‌باشد. در شکل ۴ خلاصه الگوریتم رمزنگاری و رمزگشایی قابل مشاهده است.

۴- تشکیل کلید رمز

روش پیشــنهادی رمزنگاری، جزو روش‌های کلید متقارن می‌باشــد. تمام اطلاعات لازم جهت عملیات رمزنگاری و رمزگشایی می‌بایست در کلید رمز قرار گیرد، تا عملیات در هر دو بخش به‌درستی انجام پذیرد.



شکل ۴: نتیجه حاصل از اجرای الگوریتم پیشنهادی مقاله

کلید این الگوریتم، کلیدی ۲۵۶ بیتی می‌باشــد. در جدول ۲ اجزای کلید رمز قابل مشاهده می‌باشد این اجزا عبارت‌اند از:

۱) سه بیت جهت تعیین نسخه الگوریتم استفاده می‌گردد. همان‌طورکه در قسمت‌های قبل توضیح داده شد، امکان تغییر در برخی از قسمت‌ها

مانند تعداد کاراکترهای بخش ۲-۱، تعداد کدهای مورد استفاده در کدینگ، نوع کدها و ... در نسخه‌های مختلف، وجود دارد.

۲) ۷ بیت برای متغیر n_{LB} در بخش ۲-۱

۳) ۴۰ بیت کاملاً دلخواه برای ۴ متغیر ده بیتی $\acute{A},\acute{B},\acute{C},\acute{D}$ که در بخش ۲-۲ مورد استفاده قرارگرفته است.

۴) ۱۰ بیت کاملاً دلخواه برای متغیر E' که در بخش ۲-۳ مورد استفاده قرار گرفت.

۵) ۴۰ بیت کاملاً دلخواه برای ۴ متغیر ده بیتی A_1'، B_1'، C_1' و D_1' که در بخش ۲-۴ مورد استفاده قرار گرفتند.

۶) ۳۲ بیت کاملاً دلخواه برای متغیر F که در بخش ۲-۵ و بخش ۳-۱ مورد استفاده قرار گرفت.

۷) ۶ بیت کاملاً دلخواه برای متغیر G که در بخش ۲-۵ معرفی شد.

۸) ۶ بیت کاملاً دلخواه برای متغیر H و ۶ بیت دلخواه دیگر برای متغیر I که در بخش ۳-۱ استفاده شدند.

۹) ۱۰ بیت برای متغیر $n_{added-bit}$ و ۹۶ بیت کاملاً دلخواه برای متغیر J که در بخش ۳-۲ استفاده شدند.

جدول ۲: محتویات و پارامترهای کلید رمز روش معرفی شده

عنوان	Ver	n_{lb}	Key_p	$\mathfrak{E}'$	Key_{p1}	F	G	H	I	n_{a-b}	J
تعداد بیت‌ها	3	7	40	10	40	32	6	6	6	10	96
مکان در کلید	K_{1-3}	K_{4-10}	K_{11-50}	K_{51-60}	K_{61-100}	$K_{101-132}$	$K_{133-138}$	$K_{139-144}$	$K_{145-150}$	$K_{151-160}$	$K_{161-256}$
بخش		۲-۱	۲-۲	۲-۳	۲-۴	۲-۵ ۳-۱	۲-۵	۳-۱	۳-۱	۳-۲	۳-۲

در کلید ۲۵۶ بیتی حاضر، به‌جز ۳ بیت تعیین نسخه الگوریتم و ۷ بیت متغیر n_{LB} که تابعی از تعداد کاراکتر متن آشکار می‌باشد، بقیه ۲۴۶ بیت دیگر آن کاملاً دلخواه و تصادفی می‌باشند. این بیت‌ها برای مقداردهی به متغیرهای مورد استفاده در بخش‌های الگوریتم رمزنگاری استفاده شدند. مقدار این بیت‌ها هیچ تأثیری در قدرت الگوریتم ندارند. به عبارتی کلید ضعیف و قوی در الگوریتم حاضر وجود ندارد.

می‌توان نتیجه گرفت ۲۴۶ بیت از کلید رمز(به‌جز ۳ بیت نسخه الگوریتم و ۷ بیت متغیر n_{LB}) می‌توانند هر عدد دلخواه و غیرقابل‌پیش‌بینی توسط رمزشکن باشند.

۵- بررسی نتایج الگوریتم رمزنگاری پیشنهادی

۵-۱- ارزیابی الگوریتم رمزنگاری در ارتباط با هدف اصلی پژوهش: ایجاد متن رمز شده متغیر

مهم‌ترین هدف این مقاله، معرفی الگوریتمی جهت تولید متن رمز شده متفاوت در هر بار اجرای الگوریتم، با ثابت ماندن متن آشکار و کلید رمزنگاری می‌باشد.

برای بررسی عملکرد الگوریتم در این زمینه، ۵ فایل متنی به زبان انگلیسی با طول‌های متفاوت تهیه گردید. هر کدام از این متن‌ها ۵ بار توسط کلید ثابتی رمزنگاری گشت (هر کلید کاملاً تصادفی انتخاب شد). فاصله همینگ دوبدوی این متن‌های رمزشده محاسبه گشت. ($\binom{5}{2}=10$ بار فاصله همینگ در مورد هر متن مورد استفاده قرار گرفت). در جدول ۳ مشخصات هر یک از متن‌های استفاده شده و در جدول‌های ۴ تا ۸ نتایج به‌دست‌آمده ارائه می‌گردد.

جدول ۳: معرفی متن‌های آشکار استفاده شده در آزمایش

شماره متن آشکار	طول متن آشکار (برحسب کاراکتر)	طول متن آشکار (برحسب بیت)	طول متن رمز شده (برحسب بیت)
۱	۹۲	۷۳۶	۲۰۴۸
۲	۲۱۶	۱۷۲۸	۴۰۹۶
۳	۹۰۶	۷۲۴۸	۱۴۳۳۶
۴	۳۰۷۳	۲۴۵۸۴	۴۳۰۰۸
۵	۱۰۷۵۶	۸۶۰۷۲	۱۴۵۴۰۸

جدول ۴: فاصله همینگ متن‌های رمز شده مربوط به متن آشکار۱

	I_0	I_1	I_2	I_3	I_4
I_0		۹۵۸	۹۴۲	۹۷۷	۱۰۰۳
I_1	۹۵۸		۹۵۶	۹۷۵	۹۹۳
I_2	۹۴۲	۹۵۶		۹۵۳	۱۰۰۳
I_3	۹۷۷	۹۷۵	۹۵۳		۹۸۶
I_4	۱۰۰۳	۹۹۳	۱۰۰۳	۹۸۶	

جدول ۵: فاصله همینگ متن‌های رمز شده مربوط به متن آشکار۲

	I_0	I_1	I_2	I_3	I_4
I_0		۱۹۸۳	۱۹۷۶	۲۰۱۷	۱۹۹۹
I_1	۱۹۸۳		۱۹۹۳	۱۹۸۰	۲۰۰۰
I_2	۱۹۷۶	۱۹۹۳		۲۰۴۷	۲۰۱۱
I_3	۲۰۱۷	۱۹۸۰	۲۰۴۷		۱۹۹۶
I_4	۱۹۹۹	۲۰۰۰	۲۰۱۱	۱۹۹۶	

جدول ۶: فاصله همینگ متن‌های رمز شده مربوط به متن آشکار۳

	I_0	I_1	I_2	I_3	I_4
I_0		۷۰۳۶	۷۱۰۵	۷۰۳۶	۷۰۳۹
I_1	۷۰۳۶		۷۰۶۷	۷۱۲۰	۷۰۱۳
I_2	۷۱۰۵	۷۰۶۷		۷۱۷۹	۶۸۶۲
I_3	۷۰۳۶	۷۱۲۰	۷۱۷۹		۷۱۴۵
I_4	۷۰۳۹	۷۰۱۳	۶۸۶۲	۷۱۴۵	

جدول ۷: فاصله همینگ متن‌های رمز شده مربوط به متن آشکار۴

	I_0	I_1	I_2	I_3	I_4
I_0		۲۱۳۹۱	۲۱۱۲۹	۲۱۲۱۷	۲۱۲۷۰
I_1	۲۱۳۹۱		۲۱۴۴۸	۲۱۴۱۰	۲۱۰۶۱
I_2	۲۱۱۲۹	۲۱۴۴۸		۲۱۲۸۶	۲۱۱۳۷
I_3	۲۱۲۱۷	۲۱۴۱۰	۲۱۲۸۶		۲۱۲۹۱
I_4	۲۱۲۷۰	۲۱۰۶۱	۲۱۱۳۷	۲۱۲۹۱	

جدول ۸: فاصله همینگ متن‌های رمز شده مربوط به متن آشکار۵

	I_0	I_1	I_2	I_3	I_4
I_0		۷۲۲۰۲	۷۱۹۱۷	۷۲۰۴۹	۷۱۷۹۷
I_1	۷۲۲۰۲		۷۱۹۱۷	۷۱۳۷۹	۷۱۹۹۹
I_2	۷۱۹۱۷	۷۱۹۱۷		۷۱۶۸۲	۷۱۷۳۸
I_3	۷۲۰۴۹	۷۱۳۷۹	۷۱۶۸۲		۷۱۶۵۲
I_4	۷۱۷۹۷	۷۱۹۹۹	۷۱۷۳۸	۷۱۶۵۲	

ارزیابی نتایج رمزنگاری پنج متن آشکار با طول متفاوت

هدف این پژوهش ، تولید یک روش بروز رمزنگاری، با دارا بودن استانداردهای لازم جهت رمزنگاری مدرن می‌باشد. این روش باید با استفاده از متن آشکار ثابت و ثابت ماندن کلید رمزنگاری، با هر بار اجرا، نتیجه‌ای متمایز با اجرای قبل تولید کند. در تمام آزمایش‌های انجام شده، نتایج به‌دست‌آمده گویای توانایی این الگوریتم در ایجاد این خاصیت می‌باشد. مطابق جدول ۹، در آزمایش‌های انجام شده، میانگین درصد فاصله همینگ برای فایل‌ها با طول بزرگ‌تر، کمی بیشتر از فایل‌ها با طول کمتر بود. این تمایز و فاصله همینگ برحسب درصد به میانگین ۵۰ درصد در حال همگرا شدن می‌باشد. این همگرایی ۵۰ درصد، یعنی آنکه چنانچه یک متن توسط یک کلید بارها رمزنگاری شود ، هیچ رابطه و وابستگی بین محتوای متن‌های رمز شده وجود ندارد و این تفاوت کاملاً تصادفی و با نرخ عالی ۵۰ درصد بین هر دو متن رمز شده می‌باشد.

جدول ۹: اطلاعات آماری متن‌های رمز شده یک تا پنج

شماره متن آشکار	کمترین فاصله همینگ (برحسب درصد)	بیشترین فاصله همینگ (برحسب درصد)	میانگین فاصله همینگ (برحسب درصد)
۱	۴۵/۹۹	۴۸/۹۷	۴۷/۵۸
۲	۴۸/۲۴	۴۹/۹۷	۴۸/۸۳
۳	۴۷/۸۶	۵۰/۰۷	۴۹/۲۴
۴	۴۸/۹۶	۴۹/۸۶	۴۹/۴۴
۵	۴۹/۰۸	۴۹/۶۵	۴۹/۴۶

ارزیابی زمان اجرای الگوریتم رمزنگاری و رمزگشایی پنج آزمایش متن آشکار با طول متفاوت

تمام الگوریتم رمزنگاری و رمزگشایی در محیط C# نوشته شده است و از یک کامپیوتر معمولی با هسته پردازشی ۲،۲ گیگاهرتز و حافظه رم ۲ گیگابایت برای اجرا استفاده شده است.

زمان ثبت‌شده برای اجرای متن‌های آزمایش تقریباً تناسب خطی با طول متن ورودی دارد. برای متن‌های آشکار شماره سه تا پنج، نسبت زمان پاسخگویی به طول ورودی متن آشکار تقریباً ثابت می‌باشد که نشان از خطی شدن نسبت زمان اجرا به طول ورودی می‌باشد. در متن آشکار یک و دو این نسبت کمی بالاتر از آزمایش متن آشکار سه تا پنج می‌باشد. علت این تغییر کوچک، آماده‌سازی توابعی می‌باشد که در بدو اجرای الگوریتم می‌بایست آماده شوند. زمان آماده‌سازی این توابع همچون جدول‌های جایگشت ۱۲۸ و ۱۰۲۴ تایی و جداول جانشینی و ...، به طول ورودی بستگی ندارد. پرحجم‌ترین قسمت محاسبات در عمل رمزنگاری، اضافه کردن خطاهای عمدی در قسمت کدینگ

می‌با شد. زیرا با ا ضافه کردن هر بیت خطا می‌بای ست ارزیابی شود که این خطا در زمان دی‌کدینگ در عملیات رمزگ شایی حتماً قابل ا صلاح با شد. با افزایش طول ورودی متن آ شکار، زمان آماده سازی توابع اولیه در مقابل زمان اجرای عملیات رمزنگاری قابل صـــرف‌نظر می‌گردد. به همین دلیل با افزایش طول ورودی، ز مان اجرای الگوریتم رمزنگاری تابع خطی از طول ورودی می‌گردد.

۲-۵- بررسی توانایی الگوریتم در برابر تغییرات متن آشکار و کلید و تاثیرآن در متن رمز شده

یکی از بررســی‌هایی که در روش‌های رمزنگاری صـورت می‌گیرد، ایجاد تغییر در بیت‌های کلید رمز و متن آشـکار و بررســی میزان تغییرات بر روی متن رمز شده می‌باشد. طبق ا صول کریشهف، الگوریتم رمزنگاری مناســب الگوریتمی اســت که چنانچه تغییر کوچکی حتی در حد یک بیت در متن آشــکار یا کلید ایجاد شــد، تغییرات عمده‌ای در متن رمز شـــده ایجاد شـــود. این عمل توانایی الگوریتم را در مقابله با حملات افزایش می‌دهد.

در روش پیشـــنهادی، به علت تغییر متن رمز شـــده در هر بار اجرا، بدون تغییر متن آشـــکار و کلید، این تحلیل امکان‌پذیر نمی‌باشـــد. همان‌طورکه گفته شـــد، با کلید و متن آشـــکار ثابت، حداقل فاصـــله همینگ دو متن رمز شـده ، ۴۶ درصـد می‌باشـد. به همین جهت برای بررسی تأثیر تغییرکلید بر روی متن رمز شده به‌صورت زیر عمل شد.

ازآنجـایی‌کـه هر بخش کلیـد مربوط بـه یـک بخش از الگوریتم رمزنگاری می‌باشد، به‌صورت جداگانه در بخش‌های مختلف کلید، تغییر ایجاد شد و نتایج این تغییر بر روی عملکرد آن بخش ارزیابی گشت.

۵-۲-۱- تغییر کلید در بخش $key_p(A',B',C',D')$ و $key_{p1}(A'_1,B'_1,C'_1,D'_1)$

این دو کلید که هر کدام به طول ۴۰ بیت می‌با شند، در تابع جایگ شت F_P مورد استفاده قرار می‌گیرند. برای بررسی تأثیر تغییر کلید key_p در این بخش، ابتدا بلوکی ۱۲۸ خانه‌ای بنام $Test_{128}$ که در هر خانه آن شماره آن قرارگرفته است (مطابق جدول ۱۰) ایجاد می‌گردد.

جدول ۱۰: تولید بلوک $Test_{128}$ برای انجام آزمایش تغییر بیت

شماره خانه بلوک	۰	۱	...	۱۲۷
محتویات خانه	۰	۱		۱۲۷

طبق یک کلید ۴۰ بیتی تصـــادفی بنام key1 جایگشـــت خانه‌های بلوک $Test_{128}$ محاســـبه گشـــت و در بلوک ۱۲۸ بیتی $Outtest1_{128}$ قرار گرفت. چهار بار و هر بار به‌طور تصـــادفی، بیتی از کلید key1 انتخاب و عوض شـــد و طبق کلید جدید، مجدد جایگشـــت خانه‌های بلوک $Test_{128}$ محاسبه شد. نتایج حاصل به ترتیب در $Outtest2_{128}$ و ... و $Outtest5_{128}$ قرار گرفت. شباهت دوبدوی نتایج محا سبه شد. منظور از مقدار شباهت، تعداد خانه‌هایی می‌با شد که با شماره یکسان دارای محتوای یکسان می‌باشند. نتیجه آزمایش نشان داد شبیه‌ترین دو بلوک خروجی دارای ۲۲ خانه یکسان و کمترین شباهت دارای ۸ خانه یکسان است. این نتایج در جدول ۱۱ قابل مشاهده می‌باشد.

این نتیجه، گویای تغییر عمده در نتیجه بخش جایگشت، برحسب تغییر کلید مربو طه می باشـــد. نتیجه بررســـی key_{p1} که مربوط به‌جایگشت ۱۰۲۴ خانه می‌باشد، نتیجه مشابهی ایجادکرد.

جدول ۱۱: ایجاد تغییر در بیت‌های Key_p و بررسی تأثیر آن بر بلوک خروجی جایگشت

	$O1_{128}$	$O2_{128}$	$O3_{128}$	$O4_{128}$	$O5_{128}$
$O1_{128}$		۱۲	۱۵	۲۱	۱۹
$O2_{128}$	۱۲		۸	۱۳	۱۷
$O3_{128}$	۱۵	۸		۱۹	۱۶
$O4_{128}$	۲۱	۱۳	۱۹		۲۲
$O5_{128}$	۱۹	۱۷	۱۶	۲۲	

۵-۲-۲- تغییر کلید در بخش E'

کلید E' در قسمت جابجایی کاراکترها در بخش ۲-۳ کاربرد دارد. برای بررســـی تأثیر تغییر کلید E' در این بخش، ابتدا عدد ۱۰ بیتی E' به‌تصـادف انتخاب شـد.($E = E' + 31$) میزان تغییر کاراکتر خانه k ام طبق بخش دو محاســـبه گشـــت و مقدار آن در خانه k ام بلوکی ۱۲۸ بیتی بنام $Outtestch1_{128}$ قرار گرفت. سه بار و هر بار یکی از بیت‌های E' به‌طور تصـــادفی انتخاب و عوض شـــد. طبق این تغییر بلوک‌های $Outtestch2_{128}$ و ... و $Outtestch4_{128}$ محاسبه شد. میزان شباهت دوبدوی این نتایج باهم به د ست آمد. در شبیه‌ترین حالت تنها ۷ خانه دارای محتوای یکسان در بین خانه‌های بلوک خروجی وجود داشت.

۵-۲-۳- تغییر کلید در بخش ۳۲ بیتی F و ۹۶ بیتی J

این دو بخش کلید، جهت عملیات Xor بکار برده شـــده‌اند. در آزمایش‌های انجام شده مشخص گردید که میزان تغییر برحسب درصد در این بخش، به‌طور مشابه به خروجی منتقل می‌گردد. البته این نتیجه کاملاً قابل پیش‌بینی می‌باشد.

۵-۲-۴- تغییر کلید در بخش H و I

این دو متغیر در بخش کدینک مورد استفاده قرار می‌گیرند. ازآنجایی‌که بردارهای خطای اضافه شده به‌صورت تصادفی می‌باشند و با هر بار اجرا نتایج متفاوتی ایجاد می‌شـــود، امکان بررســـی میزان تغییر خروجی برحسب تغییر کلید وجود ندارد.

۳-۵- بررسی توانایی الگوریتم در برابر حملات متداول

رمزنگاری موفق، روشی است که در مقابل حملات جهت شکستن آن مقاوم باشد. به همین جهت برای بررسی کارایی الگوریتم پیشنهادی، مقاومت آن در برابر حملات متداول مورد ارزیابی قرار گرفت.

۱-۳-۵- تحلیل خطی رمز

تحلیل خطی رمز با فرض در اختیار داشتن حجم بزرگی از بلوک‌های رمز شده و معادل رمز نشده آن‌ها انجام می‌گیرد. تحلیل‌گر سعی می‌کند بین بیت‌های متن ورودی، متن رمز شده و کلید اصلی رابطه‌ای خطی پیدا کند. اساس این روش که توسط میتزورو ماتسویی مطرح شده است[۱۰]، تقریب خطی جدول‌های ثابت P-Box و S-Box می‌باشد.

S-Box ها غیر خطی‌ترین عناصر یک الگوریتم رمزنگاری هستند. تحلیل‌گر باید ترکیب‌های مختلف ورودی و خروجی این S-Box ها را بررسی نماید و به کمک علم آمار و احتمال، تقریب خطی از آن ارائه نماید. برای انجام این تقریب، تحلیل‌گر باید از ورودی سیستم رمزنگاری شروع کند و تا رسیدن به دور ماقبل آخر، تمام مسیرهایی که ورودی‌های مورد نظر بر روی آن‌ها تأثیرگذارند را شناسایی کند. سپس P-Box و S-Boxهای این مسیرها تقریب خطی زده شود و به کمک بلوک‌های متن رمز شده و رمز نشده، تقریب را به واقعیت نزدیک کند و کلید و یا قسمتی از آن را کشف نماید. برای مصون ماندن از حمله تحلیل خطی، S-Boxها باید به‌گونه‌ای انتخاب شوند که روابط تقریب خطی آن‌ها کمترین بایاس ممکن را داشته باشد.

در الگوریتم پیشنهادی، جدول‌های P-Box تابعی از کلید رمز می‌باشند. همچنین جانشینی کاراکترهای متن ورودی با کاراکترهای جدید(عملیاتی مشابه اپراتور S-Box) نیز تابعی از کلید رمز می‌باشد. به همین جهت، انجام تقریب آن‌ها غیرممکن می‌گردد و حمله خطی رمز در این روش بلااثر می‌گردد.

علاوه بر وابستگی این دو اپراتور به کلید رمز، خروجی این الگوریتم با ثابت ماندن کلید رمز، نسبت به اجرای قبل متمایز می‌باشد. بنابراین حتی اگر جدول‌های P-Box و S-Box نیز ثابت بودند، امکان انجام این تقریب و حمله از طریق آن وجود نداشت.

۲-۳-۵- تحلیل تفاضلی رمز

این روش بر این اصل استوار است که تغییرات بین دو بلوک از متن اصلی آشکار (حتی به‌اندازه یک بیت) چگونه بر روی خروجی متن رمز شده تأثیر می‌گذارد و نتیجه رمز شده این دو بلوک به چه اندازه با یکدیگر اختلاف دارند[۱۳-۱۱]. این رابطه می‌تواند در شرایط خاص، برخی از بیت‌های کلید را آشکار کند. به‌طورمعمول میزان اختلاف، معیار فاصله همینگ دو الگوی بیتی در نظر گرفته می‌شود. عملیاتی مانند جایگشت تحت جدول ثابت، و عمل Xor هیچ مقاومتی در برابر تحلیل تفاضلی ندارند. در این روش تمام تغییرات ممکن به جدول‌های جانشینی اعمال می‌گردد و تغییرات خروجی ثبت می‌شود. با تحلیل آماری این نتایج و تشکیل جدول توزیع اختلاف، شاخص‌های مؤثر برای حمله به کل سیستم رمز استخراج می‌گردد. در ادامه با استفاده از تعداد کافی متن رمز شده و معادل رمز نشده آن‌ها، طی مراحلی کلید واقعی محاسبه می‌گردد.

نقطه قوت الگوریتم پیشنهادی که باعث مقاومت کامل در برابر این حمله می‌شود آن است که جدول‌های جایگشت و جانشینی ثابت نمی‌باشند. مهم‌تر از آن با ثابت ماندن کلید رمز، خروجی این الگوریتم نسبت به اجرای قبل متفاوت می‌باشد. در عمل نمی‌توان با تغییر ورودی به‌اندازه Δx، مقدار تغییر خروجی(Δy) را محاسبه کرد. زیرا حتی اگر تغییری در ورودی ایجاد نشود$(\Delta x = 0)$، خروجی به میزان قابل‌توجهی به‌صورت تصادفی تغییر می‌کند. این خاصیت باعث مقاومت کامل روش پیشنهادی در برابر تحلیل تفاضلی رمز می‌گردد[۱۴].

۳-۳-۵- آزمون جامع فضای کلید

همان‌طورکه گفته شد، روش پیشنهادی در برابر دو روش مشهور و کلاسیک از حملات علیه سیستم‌های رمز متقارن، یعنی تحلیل خطی رمز و تحلیل تفاضلی رمز مقاوم می‌باشد. بنابراین هیچ قسمتی از کلید توسط این دو روش قابل استخراج نیست. تحلیل‌گر به‌ناچار می‌بایست تمام حالات کلید رمز را برای شکستن کلید امتحان نماید[۱۵]. همان‌طورکه در بخش تشکیل کلید رمز اشاره شد، تعداد ۲۴۶ بیت از بیت‌های کلید رمز، کاملاً تصادفی انتخاب می شود. هیچ کلید رمزی از نظر ضعف و قدرت با رمز دیگر تفاوتی ندارد. بنابراین امکان وجود 2^{246} کلید متمایز رمز وجود دارد. در حال حاضر با تکنولوژی موجود، کلیدهایی که بیش از 2^{90} حالت مستقل داشته باشند، در مقابل آزمون جامع فضای کلید ایمن می‌باشند.

۴-۳-۵- سایر حمله‌ها

در سایر روش‌های حمله به متن رمز شده، در اکثر موارد (از جمله حمله مکمل یا حمله از طریق ویژگی بسته بودن و ...)، تحلیل‌گر متن‌های منتخبی که وابستگی‌های خاصی باهم دارند را به ورودی سیستم رمزنگار، اعمال می‌کند [۱۶]. تحلیل‌گر به کمک تحلیل خروجی سیستم رمزنگار و نوع وابستگی‌های ورودی، سعی در کشف بخشی از کلید، برای کاهش بیت‌های کلید رمز، برای حمله آزمون جامع دارد[۱۷]. همان‌طورکه قبلاً ذکر شد خروجی این الگوریتم با ثابت ماندن کلید رمز و متن آشکار، نسبت به اجرای قبل متفاوت می‌باشد. بنابراین امکان چنین حملاتی به این روش وجود ندارد.

۴-۵- مقایسه الگوریتم رمزنگاری پیشنهادی با سایر الگوریتم‌های هم الگو

همان‌طورکه در بخش مقدمه مقاله گفته شد، ترکیب رمزنگاری و تئوری کدینگ، در سامانه‌های توأم رمزنگاری و کدگذاری کانال مورد

استفاده قرار می‌گیرد. در این روش‌ها، برای اضافه نمودن خطاهای عمدی از روش‌های کدینگ استفاده می شود و همین امر باعث افزایش طول متن رمزنگاری نسبت به متن آشکار می‌گردد.

توانمندی هر الگوریتم در برابر حملات، مهم‌ترین خاصیت یک روش رمزنگاری می‌باشد. در بخش قبل، توانمندی الگوریتم پیشنهادی در برابر حملات متداول رمزنگاری ارزیابی گشت. در بخش حاضر، ویژگی‌های الگوریتم پیشنهادی از نظر طول کلید، حداقل فاصله همینگ و نرخ افزایش طول متن رمزنگاری نسبت به طول متن آشکار نسبت به الگوریتم‌های هم نوع، مورد ارزیابی قرارمی‌گیرد.

از روش‌های معروف در سامانه‌های توأم رمزنگاری و کدگذاری کانال می‌توان از روش رائو، روش رائو-نام، روش استریک-تیلبورگ، روش سان-شیه، روش باربرو-ایتروس و روش باربرو-تنا در دو دهه گذشته نام برد. همچنین در چند سال اخیر این سامانه مجدد مورد توجه قرارگرفته و روش‌های جدیدی همچون روش صبحی-افشار، روش هوشمند، روش اسماعیلی، روش اسماعیلی-گیلور معرفی گشتند. در زیر سامانه پیشنهادی، با نتایج این روش‌ها مقایسه می‌شود[۱۸].

طول کلید

در جدول ۱۲ نتیجه مقایسه طول کلید روش پیشنهادی با روش‌های نامبرده مشاهده می‌شود.

جدول ۱۲: مقایسه طول کلید روش‌های رمزنگاری مبتنی بر کدینگ

سامانه رمزنگاری	$C(n,k)$	طول کلید
رائو[۱۹]	$C(1024,524)$	۲ مگابیت
رائو-نام[۲۰]	$C(72,64)$	۱۸ کیلوبیت
استریک-تیلبورگ[۲۱]	$C(72,64)$	۱۸ کیلوبیت
سان و شیه[۲۲]	$C(49,36)\ over\ F_2^3$	۴۲ کیلوبیت
باربرو و ایتروس[۲۳]	$C(30,20)\ over\ F_2^8$	۴/۹ کیلوبیت
باربرو و تنا[۲۴]	$C_1(10,9), C_2(10,3)\ over\ F_2^4$	۳/۸۷ کیلوبیت
صبحی و افشار[۲۵]	$C(2044,1024)$	۲/۵ کیلوبیت
هوشمند[۲۶]	$C(2470,2223)$	۳/۵۵ کیلوبیت
اسماعیلی[۲۷]	$C(2048,1536)$	۲/۱۹۱ کیلوبیت
اسماعیلی و گیلور[۲۸]	$C(2048,1536)$	۲/۲۲ کیلوبیت
هوشمند[۲۹]	$C(1024,832)$	۵ کیلوبیت
سامانه پیشنهادی	از ۴ کد استفاده شد که میانگین آن‌ها به‌صورت زیر است: $C(60,36)$	۲۵۶ بیت

همان‌طورکه قابل مشاهده است، طول کلید رمزنگاری سامانه پیشنهادی در مقایسه با روش‌های هم نوع، کاهش بسیار چشمگیری داشته است. این کاهش در بحث انتقال کلید رمز، در روش‌های کلید رمزنگاری متقارن بسیار مهم و حائز اهمیت می‌باشد.

حداقل فاصله همینگ

همان‌طورکه در بخش کدینگ گفته شد، اگر حداقل فاصله همینگ روش کدگذاری d_{min} در نظر گرفته شود، در انتقال داده اگر حداکثر s بیت دچار خطا شود $(s = d_{min} - 1)$، در گیرنده امکان کشف قطعی خطا وجود دارد. همچنین اگر تعداد بیت‌های خطا کوچک‌تر یا مساوی آستانه t باشد $(t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor)$، گیرنده علاوه بر کشف خطا قادر است خطای رخ داده را به صورت قطعی تصحیح نماید و کد اصلی را کشف کند. البته روش کدگذاری امکان کشف و تصحیح خطای بیشتر را ممکن است داشته باشد. این موضوع قطعی نمی‌باشد و حتماً به‌صورت موردی می‌بایست بررسی گردد.

در بخش کدینگ روش پیشنهادی، آن‌قدر خطای عمدی به بردار کد شده اضافه شد تا جایی که بتوان در زمان رمزگشایی این خطاها را کشف و تصحیح کرد.

در بخش اضافه نمودن خطا در الگوریتم پیشنهادی، یک نوآوری مورد استفاده قرار گرفت. بردار خطایی با نام $u_{extra-error}$ طبق الگوریتمی بر اساس کلید رمز و نتایج رمزگذاری قبلی در هر مرحله محاسبه گشت. طول این خطا دقیقاً به طول بردار کد شده بود. همین امر سبب ایجاد خاصیت تغییر نزدیک به ۵۰ درصد، نسبت به اجراهای قبل در این روش شد.

در مقایسه با سایر روش‌های ذکر شده، استفاده از این نوآوری، باعث ایجاد تغییری کاملاً منحصربه‌فرد شد. نتیجه حاصل، نشان‌دهنده توانایی الگوریتم در تولید خطای عمدی گسترده در متن رمز شده می‌باشد.

نرخ کد

در روش‌های ذکر شده، برای ایجاد خطای عمدی، از کدینگ استفاده می‌شود. در روش‌های کدینگ، برای ایجاد قابلیت تصحیح خطا می‌بایست بیت‌های اضافه به بردار اولیه اضافه شود. در کد $C(n,k)$ نرخ کد به‌صورت $R = k/n$ تعریف می‌شود. در جدول ۱۳ نتیجه مقایسه نرخ کد روش پیشنهادی با سایر روش‌ها قابل مشاهده است.

جدول ۱۳: مقایسه نرخ کد روش‌های رمزنگاری مبتنی بر کدینگ

سامانه رمزنگاری	$C(n,k)$	نرخ کد
رائو[۱۹]	$C(1024,524)$	۰/۵۱
رائو-نام[۲۰]	$C(72,64)$	۰/۸۹
استریک-تیلبورگ[۲۱]	$C(72,64)$	۰/۸۹
سان و شیه[۲۲]	$C(49,36)\ over\ F_2^3$	۰/۷۳
باربرو و ایتروس[۲۳]	$C(30,20)\ over\ F_2^8$	۰/۶۶
باربرو و تنا[۲۴]	$C_1(10,9), C_2(10,3)\ over\ F_2^4$	۰/۶
صبحی و افشار[۲۵]	$C(2044,1024)$	۰/۵
هوشمند[۲۶]	$C(2470,2223)$	۰/۹
اسماعیلی[۲۷]	$C(2048,1536)$	۰/۷۵
اسماعیلی و گیلور[۲۸]	$C(2048,1536)$	۰/۷۵
هوشمند[۲۹]	$C(1024,832)$	۰/۸۱
سامانه پیشنهادی	از ۴ کد استفاده شد که میانگین آن‌ها به‌صورت زیر است: $C(60,36)$	۰/۵۹

همان‌طورکه ملاحظه می‌شود این روش در زمینه نرخ کد، عملکرد متوسطی از خود نشان می‌دهد.

۵-۵- تضمین امنیت کامل سامانه رمزنگاری معرفی شده

کلود شانون اثبات کرد که هرگاه رمزنگاری اطلاعات از طریق XOR کردن متن و کلید رمز شرایط زیر را داشته باشد «بی قید و شرط امن» خواهد بود و میزان توان محاسباتی و هوش رمزشکن هیچ تأثیری در شکستن رمز نخواهد داشت:

شرط اول: بیت‌های کلید به‌صورت کاملاً تصادفی انتخاب شود و احتمال صفر و یک بودن بیت‌های کلید دقیقاً ۰/۵ و مستقل از یکدیگر باشند.

شرط دوم: طول کلید و متن اصلی برابر باشند؛ شرط دوم از شرط اول قابل استنتاج است چرا که هرگاه طول کلید از طول متن کوتاه‌تر باشد کلید تکرار خواهد شد که این موضوع شرط استقلال بیت‌های کلید را مخدوش می‌کند.

شرط سوم: برای رمزنگاری متون مختلف هیچ‌گاه از یک کلید دوبار استفاده نشود.

تمام شرط‌ها باعث تولید یک متن رمز شده که احتمال وقوع هر بیت صفر و یک در خروجی ۵۰ درصد باشد، می‌شود.

در پژوهش حاضر، در تمام آزمایش‌های انجام شده، نتایج به‌دست‌آمده گویای توانایی الگوریتم پیشنهادی در ایجاد متن رمز شده متمایز در هر بار اجرا، علیرغم ثابت ماندن متن آشکار و کلید رمز است. فاصله همینگ دو متن رمز شده مربوط به یک متن، به عدد ۵۰ درصد در حال همگرایی می‌باشد. در الگوریتم معرفی شده در این مقاله، متن آشکار به‌صورت بلوک‌های ۱۲۸ کاراکتری، بلوک‌بندی شد و سپس عملیات رمزنگاری شروع می‌شود. با توجه به نتیجه به‌دست‌آمده، در زمانی که متن آشکار تنها شامل یک بلوک نیز است، میانگین فاصله همینگ ۴۷/۵۸درصد و کمترین فاصله همینگ ۴۵/۹۹درصد به دست آمد. این نتیجه بسیار نتیجه مطلوب و مناسبی می‌باشد.

همگرایی ۵۰ درصدی، یعنی آنکه چنانچه یک متن توسط یک کلید بارها رمزنگاری شود ، هیچ رابطه و وابستگی بین محتوای متن‌های رمز شده وجود ندارد و احتمال وقوع هر بیت صفر و یک در خروجی رمز شده ۵۰ درصد می‌شود.

طبق مقاله کلود شنون به علت عدم وابستگی متن‌های رمز شده و تغییرات ۵۰ در صدی مربوط به یک متن و کلید ثابت ، این روش کاملاً ایمن می‌باشد و هیچ نوع حمله از حملات چهاگانه: ((حمله فقط به متن رمز شده، حمله به متن روشن معلوم، حمله به متن روشن منتخب، حمله تطبیقی به متن روشن منتخب)) نمی‌تواند خللی در امنیت آن ایجاد کند.

۶- نتیجه

در مقاله حاضر، بر پایه سامانه‌های توأم رمزنگاری وکدگذاری کانال، یک روش رمزنگاری جدید ارائه گشت. سامانه معرفی شده ضمن حفظ امنیت یک روش رمزنگاری که مهم‌ترین خواسته از این الگوریتم‌ها می‌باشد توانست به نقاط مثبت زیر در مقایسه با این روش‌ها دست پیدا کند:

- تولید الگوریتم رمزنگاری جدید به کمک علم نظریه اعداد و کدینگ به‌عنوان یک روش مستقل از کانال‌های انتقال
- افزایش فاصله همینگ چشمگیر در مقایسه با این روش‌ها

ایجاد فاصله همینگ حدود ۵۰ در صدی هر متن رمز شده با متن‌های رمز شده اجراهای قبل.

- کاهش چشمگیر طول کلید رمزنگاری (۲۵۶ بیت) در مقایسه با سامانه‌های توأم رمزنگاری وکدگذاری کانال
- نوآوری در تولید بردار خطای $u_{extra-error}$.

این بردار خطا که در بخش تولید خطای عمدی بر اساس کلید رمز و نتایج بخش‌های قبل تولید شد دارای وزن همینگ بالا می‌باشد.

در کنار نقاط مثبت بالا، این روش نقاط ضعف زیر را در مقایسه با این روش‌ها دارای می‌باشد:

- این روش در زمینه نرخ کد در مقایسه با روش‌های ترکیبی، عملکرد متوسطی از خود نشان می‌دهد.

برای بهبود این وضعیت، دو پیشنهاد برای ادامه کار وجود دارد: یکی آنکه در بخش کدینگ، ضمن حفظ فاصله همینگ،کدهای دیگری مورد آزمایش قرار گیرد. دیگر آنکه الگوریتمی جدید جهت تبدیل کاراکترها به اعداد باینری در بخش ۲-۲ طراحی شود. به‌عنوان‌مثال، این الگوریتم تبدیل کاراکتر به بیت دارای بخش فشرده‌سازی نیز باشد. با تولید الگوریتم مناسب جهت انجام این کار، می‌توان حجم داده‌ها را کاهش داد. به عبارتی قبل از بخش کدینگ، حجم داده های ورودی بخش کدینگ کاهش یابد. به این ترتیب می‌توان بخشی از نرخ افزایش حجم داده در بخش کدینگ را جبران نمود.

- این روش حجم عملیات و پردازش بالاتری نسبت به سامانه‌های هم نوع دارد.

وجود الگوریتمی با حجم عملیات پایین‌تر، از نظر کاهش زمان اجرای الگوریتم و کاهش منابع مصرفی همچون انرژی حائز اهمیت است. در این مقایسه باید نوع کاربرد این الگوریتم هم مورد توجه قرار گیرد. سامانه‌های توأم رمزنگاری وکدگذاری کانال، اصولاً در کاربردهای آنلاین مورد استفاده قرار می‌گیرند. سامانه معرفی شده یک روش مستقل رمزنگاری است که می‌تواند به‌صورت آفلاین مورد استفاده قرار گیرد.

۷- کاربردهای الگوریتم پیشنهادی و ادامه کار

امروزه طراحی P-Box ها و S-Box ها برای بهینه کردن روش های رمزنگاری [۳۰] مورد توجه هستند. برای این ابزارها، کاربردهای جدیدی نیز پیدا شده است. استفاده از این ابزارها در رمزنگاری فایل‌های تصویری به تعداد قابل‌توجه به چشم می‌خورد. در این مقاله، الگوریتمی برای تولید P-Box و S-Box پویا مورد استفاده قرار گرفت. پیشنهاد می‌شود این جعبه‌ها جانشین جعبه‌های ثابت مورد استفاده در این روش‌ها شود[۳۱]. از قابلیت ایجاد خطای عمدی در سایر کاربردها می‌توان کمک گرفت. به‌عنوان نمونه، از این الگوریتم می‌توان در پنهان

نگاری اســـتفاده کرد. پنهان نگاری به معنای قرار دادن داده‌ها در فایل یک تصـویر می‌باشـد، به‌طوری‌که به‌صـورت مخفیانه همراه آن ارسـال گردد. و فردی از وجود آن آگاه نگردد. در گیرنده این اطلاعات از تصویر استخراج و بازسازی می‌گردد. امروزه الگوریتم‌های رمزنگاری در ساختار الگوریتم‌های پنهان نگاری جایگاه ویژه‌ای دارند[۳۲].

در ادامه کار پیشـــنهاد می‌شـــود از این الگو در ســـامانه‌های توأم رمزنگاری وکدگذاری کانال نیز اســـتفاده گردد و بعد از کســـب نتیجه مطلوب، برای افزایش ســـرعت، الگوریتم آن به‌صـــورت ســـخت‌افزاری (به‌عنوان نمونه پیاده‌سازی بر روی FPGA) اجرا گردد.

پیوست یک: نحوه پیاده‌سازی الگوریتم جایگشت پویا بخش دو مقاله

همان‌طورکه در متن مقاله گفته شـــد، در هر مرحله شـــماره یکی از خانه‌های بلوک ورودی که هنوز انتخاب نشـــده اســـت براســـاس کلید انتخاب می‌گردد و محتویات آن در اولین خانه خالی بلوک خروجی قرار می‌گیرد. گام‌های زیر جهت تکمیل این فرآیند انجام می‌پذیرد:

1. آماده‌ســـازی مقادیر متغیر تابع انتخاب خانه بلوک ورودی.

در پیاده‌ســـازی این الگوریتم کاربر عدد n_1 را به‌عنوان طول بلوک ورودی و خروجی P-BOX و چهار عدد دلخواه A'، B'، C' و D' را به‌عنوان ورودی تابع معرفی می‌کند.

عـدد A برابر ۱۰۰۰ + A' و عـدد B برابر ۱۰۰۰+B' قرار می‌گیرد. (این عمل تضـــمین می‌کند حداقل مقدار A و B برابر ۱۰۰۰ می‌باشد.)

عدد C برابر ۱۰۰+C' و عدد D برابر ۱۰+D' قرار می‌گیرد. (این عمل تضـــمین می‌کند حداقل مقدار C برابر ۱۰۰ و حداقل مقدار D برابر ۱۰ است.)

عدد Pc برابر بزرگ‌ترین عدد اول که Pc<C باشـــد انتخاب می شود و بردار P به صورت معادله ۱۱ که شامل تمام اعداد اول کوچک‌تر یا مساوی Pc می‌باشد تعریف می‌گردد.

$$P = (Pc \text{ و } \dots \text{ و } ۵ ،۳ ،۲) \quad (۱۱)$$

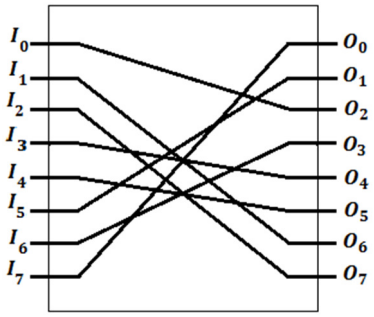
2. عدد اول n که کوچک‌ترین عدد اولی می‌با شد که $n_1 \le n$ ا ست، انتخاب می‌شود.
3. عدد m برابر تعداد خانه های بلوک ورودی که هنوز برای قرار گرفتن در بلوک خروجی انتخاب شده‌اند تعریف می شود. در ابتدا اجرا $m = n_1$ می‌باشـــد و بعد از هر مرحله یک واحد از m کم می‌شود.
4. انتخاب شماره خانه بلوک ورودی و انتقال به بلوک خروجی.

در هر مرحله شماره خانه انتخابی بلوک ورودی برای قرارگیری در بلوک خروجی به‌صورت زیر محاسبه می‌شود:

$$\left[\left((A - P[k \bmod D])^{(B+P[k \bmod D])}\right) \bmod n\right] \bmod m \quad (۱۳)$$

که P[K] برابر مقدار مؤلفه k ام بردار P می‌باشد.

مثال: اگر طول بلوک ورودی برابر ۸ باشد ($\boldsymbol{n_1 = 8}$) و برای جایگشت خانه‌های این بلوک، از الگوریتم معرفی شده به ازای پارامترهای (A'=200, B'=170, C'=90, D'=20) استفاده شود، نتایج به‌صورت جدول ۱۴ در ۸ مرحله اجرای الگوریتم به دست می‌آید. شکل ۵ جعبه‌جایگشت این مثال را نشان می‌دهد.



شکل ۵: نمایش جعبه‌جایگشت مثال پیوست یک

جدول ۱۴: محاسبات جعبه‌جایگشت مثال پیوست یک در ۸ مرحله

شماره خانه انتخابی $= \left[\left((A - P[k \bmod D])^{(B+P[k \bmod D])}\right) \bmod n\right] \bmod m$, n=11

مرحله	m	شماره خانه انتخابی	Input Block								Output Block							
0			Home Number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	0	1	2	3	4	5	6	7	Contain								
1	8	7	Home number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	0	1	2	3	4	5	6	7	Contain	7							
2	7	5	Home Number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	0	1	2	3	4	5	6		Contain	7	5						
3	6	0	Home number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	0	1	2	3	4	6			Contain	7	5	0					
4	5	4	Home Number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	1	2	3	4	6				Contain	7	5	0	6				
5	4	2	Home number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	1	2	3	4					Contain	7	5	0	6	3			
6	3	2	Home Number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	1	2	4						Contain	7	5	0	6	3	4		
7	2	0	Home number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	1	2							Contain	7	5	0	6	3	4	1	
8	1	0	Home Number	0	1	2	3	4	5	6	7	Home Number	0	1	2	3	4	5	6	7
			Contain	2								Contain	7	5	0	6	3	4	1	2

پیوست دو

اثبات: اگر عدد m به صورت $m = G + 0.9$ تعریف گردد$(G \in N)$، عدد $\sqrt{m}$ عددی گنگ است.(استفاده در بخش ۲-۵)

برهان خلف: عدد $\sqrt{m}$ عددی گویا است، در نتیجه:

$$c = \sqrt{m} \in Q \rightarrow c = \frac{a}{b} \; ; \; (a,b) = 1 \; ; a,b \in N$$

$$\left.\begin{matrix} \mathrm{m} = \mathrm{c}^2 \\ \mathrm{m} = \mathrm{G} + 0.9 \; , \mathrm{G} \in \mathrm{N} \end{matrix}\right\} \rightarrow \mathrm{G} + \frac{9}{10} = \frac{\mathrm{a}^2}{\mathrm{b}^2}$$

$$\rightarrow \begin{cases} 10\mathrm{G} + 9 = \dfrac{10\mathrm{a}^2}{\mathrm{b}^2} \\ 10\mathrm{G} + 9 \in \mathrm{N} \end{cases}$$

$$\left.\begin{matrix} \rightarrow \dfrac{10a^2}{b^2} \in N \\ (a,b) = 1 \end{matrix}\right\} \rightarrow b^2 | 10 \rightarrow b = 1 \rightarrow 10G + 9 = 10a^2$$

$$9 = 10(a^2 - G) \rightarrow 10|9 \text{ تناقض} \rightarrow \sqrt{m} \in Q'$$

مراجع

[1] C. Shannon, “Communication theory of secrecy system,” Bell System Technology Journal, vol. 28, pp. 656–715, 1990.

[2] K. M. Martin and D. R. Stinson, “Error decodable secret sharing and one-round perfectly secure message transmission for general adversary structures,” Cryptography and Communications - Discrete Structures, Boolean Functions and Sequences, vol. 3, pp. 65-86, 2011.

[3] W. Meier and O. Staffelbach, “Fast correlations attacks on certain stream ciphers,” Journal of Cryptology, Springer, pp. 159-176, 1989.

[4] M. Zobeiri and B. Mazloom-Nezhad Maybodi, “Introducing dynamic P-Box and S-Box based on modular calculation and key encryption for adding to current cryptographic systems against the linear and differential cryptanalysis,” ARPN Journal of Engineering and Applied Sciences, vol. 12, pp. 856-862, 2017.

[5] T. Siegenthaler, “Correlation-immunity of nonlinear combining functions for cryptographic applications,” IEEE Transactions on Information Theory IT, pp. 776–780, 1984.

[6] M. Grangetto, E. Magli and G. Olmo, “Multimedia selective encryption by means of randomized arithmetic coding,” IEEE Transactions on Multimedia, vol. 8, pp. 905–917 2006.

[7] C. Monico, J. Rosenthal and A. Shokrollahi, “Using low density parity check codes in the McEliece cryptosystem,” Proc. IEEE International Symposium Information Theory, Italy, 2000.

[8] M. Baldi, F. Chiaraluce, R. Garello and F. Mininni, “Quasi-cyclic LDPC codes in the McEliece cryptosystem,” Proc. IEEE International Conference Communications, Glasgow, UK, pp. 951–956, 2007.

[9] M. Baldi and F. Chiaraluce, “Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes,” Proc. IEEE International Symposium Information Theory, Nice, France, pp. 2591–2595, 2007.

[10] M. Matsui, “Linear cryptanalysis method for DES cipher, ” Advances in Cryptology EUROCRYPT, Springer-Verlag, 1994.

[11] Biham and Shamir, “Differential cryptanalysis of DES-like cryptosystems,” Technical Report CS90-16, 1990.

[12] C. H. Kim, “Differential fault analysis of AES: toward Reducing number of faults,” Journal of Information Sciences, vol. 199, pp. 43–57, 2012.

[13] J. Kim, S. Hong and J. Lim, “Impossible differential cryptanalysis using matrix method,” Discrete Mathematics, pp. 988–1002, 2010.

[14] K. Sakiyama, Y. Li, M. Iwamoto and K. Ohta, “Information-theoretic approach to optimal differential fault analsysis,” IEEE Transactions on Information Forensics and Security, vol. 7, pp. 109-120, 2012.

[15] D. Coppersmith, “The data encryption standard (DES) and its strength against attacks,” IBM Journal of Research and Development, vol. 38, 1994.

[16] شهرام جمالی، عرفان آقایی کیاسرایی، « بهبود حمله مکعبی کانال جانبی بر روی الگوریتم‌های بلوکی »، مجله مهندسی برق دانشگاه تبریز، دوره ۴۵، شماره ۴، صفحه ۷۸-۶۹، ۱۳۹۴.

[17] P. Xu and H. Jin, “Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack,” IEEE Transactions on Computer, vol. 62, pp. 2266–2278, 2013.

[18] M. Esmaeili and T. A. Gulliver, “A secure code based cryptosystem via random insertions, deletions, and errors,” IEEE Communications, vol. 20, pp. 870–873, 2016.

[19] T. N. R. Rao, “Joint encryption and error correction schemes,” Proc. Int. Symp.Computer Architecture, pp. 240–241, 1984.

[20] T. R. N. Rao and K. H. Nam, “Private-key Algebraic-code encryption,” IEEE Transactions on IT, vol. 4, pp. 829-833,1987.

[21] R. Struik, J. Tilburg, “The Rao–Nam scheme is insecure against a chosen plaintext attack”, Proc. CRYPTO, pp. 445–457, 1988.

[22] H. M. Sun, S. P. Shieh, ‘On private-key cryptosystems based on product codes,” Proc. 3rd Australasian Conference Information Security and Privacy, pp. 68–79, 1998.

[23] A. I. Barbero and Ytrehus, “Modifications of the Rao–Nam cryptosystem,” Proc. International Conference Coding Theory Cryptography and Related Areas, pp. 1–13, 1998.

[24] A. I. Barbero, J. G. Tena, “A Rao–Nam like cryptosystem with product codes,” Proc. 6th International Conference Finite Fields and Applications on Coding Theory, pp. 22–36, 2001.

[25] A. Sobhi Afshar, T. Eghlidos and M. R. Aref, “Efficient secure channel coding based on quasi-cyclic low-density parity check codes,” IET Communication, vol. 3, pp. 279–292, 2009.

[26] R. Hooshmand, T. Eghlidos and M. R. Aref, “Improving the Rao–Nam secret key cryptosystem using regular EDF-QC-LDPC codes,” ISeCure, vol. 3, pp. 3–14, 2012.

[27] M. Esmaeili, M. Dakhilalian and T. A. Gulliver, "New secure channel coding scheme based on randomly punctured quasi-cyclic low-density parity check codes," IET Communication, vol. 8, pp. 2556–2562, 2014.

[28] M. Esmaeili and T. A. Gulliver "Joint channel coding-cryptography based on random insertions and deletions in QC-LDPC codes," IET Communication, vol. 9, pp. 1555–1560, 2015.

[29] R. Hooshmand, M. R. Aref and T. Eghlidos, "Secret key cryptosystem based on non-systematic polar codes," Wirel. Pers. Communication, vol. 84, pp. 1345–1373, 2015.

[30] پرهام درّی، علی قیاسیان و حسین سعیدی، «طراحی و پیاده‌سازی رمزنگار AES در بستر FPGA برای خطوط پرسرعت»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۴، شماره ۱، صفحه ۱۶۷-۱۵۳، ۱۳۹۵.

[31] L. Q. Liu Z, J. Dai and A. M. Sun X, "A new kind of double image encryption by using a cutting spectrum in the 1-D fractional Fourier transform domains," Opt. Express, vol. 282, pp. 1536-1540, 2009.

[32] K. Challita and H. Farhat, "Combining steganography and cryptography: new directions," International Journal on New Computer Architectures and Their Applications, vol. 1, pp. 199-208, 2011.