ارائه یک طرح رمزنگاری امن مبتنی بر کدهای قالبی خطی

تاریخ دریافت:
۳ اردیبهشت ۱۴۰۲

تاریخ پذیرش:
۸ مهرماه ۱۴۰۲

رضا هوشمند*[۱]
۱. استادیار دانشکده مهندسی برق، دانشگاه علوم و فنون هوایی شهید ستاری

چکیده
در این مقاله، یک طرح لفافه‌بندی ترکیبی مبتنی بر کدهای قطبی ارائه شده است که در آن، مزایا و کارآیی سیستم‌های رمزنگاری نامتقارن و متقارن با یکدیگر ترکیب شده‌اند. طرح لفافه‌بندی ترکیبی پیشنهادی شامل دو بخش است: الف) لفافه‌بندی کلید مبتنی بر کدهای قطبی که وظیفه آن لفافه‌گذاری، ارسال و لفافه‌گشایی کلید مخفی مورد نیاز در طرح لفافه‌بندی داده مبتنی بر کدهای قطبی است، ب) لفافه‌بندی داده مبتنی بر کدهای قطبی که وظیفه آن لفافه‌گذاری، ارسال و لفافه‌گشایی داده بین طرفین مجاز با استفاده از کلید مخفی رد و بدل شده به وسیله طرح لفافه‌بندی کلید است. تحلیل امنیتی طرح لفافه‌بندی ترکیبی پیشنهادی نیز با اعمال حمله جستجوی فراگیر کلید و حمله کدگشایی مجموعه اطلاعات انجام شده است. نتایج تحلیل امنیت نشان می‌دهد که طرح پیشنهادی در برابر حملات مورد نظر امن است. سطح کارآیی طرح پیشنهادی نیز بر حسب نرخ شکست رمزگشایی و طول کلید مورد ارزیابی قرار گرفته است. نتایج تحلیل نیز بر مناسب بودن نرخ شکست رمزگشایی و کوتاه بودن طول کلید دلالت دارد.

واژه‌های کلیدی: رمزنگاری پساکوانتوم، لفافه بندی، کدهای قطبی

Introducing a secure encryption scheme based on linear block codes

Abstract
In this article, a hybrid encapsulation scheme based on polar codes is presented, in which the advantages and efficiency of asymmetric and symmetric encryption systems are combined with each other. The proposed hybrid encapsulation scheme includes two parts: a) key encapsulation based on polar codes whose task is to encapsulate, send and de-envelop the secret key required in the data encapsulation scheme based on polar codes, b.) data encapsulation based on polar codes whose task is to encapsulate, send and deencapsulate data between authorized parties using the secret key exchanged by the key encapsulation scheme. The security analysis of the proposed hybrid encapsulation scheme has also been done by applying the comprehensive key search attack and the information set decoding attack. The results of the security analysis show that the proposed plan is safe against the intended attacks. The efficiency level of the proposed scheme has also been evaluated in terms of decryption failure rate and key length. The results of the analysis also indicate the suitability of the decoding failure rate and the short length of the key.

Key words: Post-quantum cryptography, encapsulation, polar codes

ایمیل نویسنده مسئول: rhooshmand50@yahoo.com

۱-مقدمه

با ظهور کامپیوترهای کوانتومی و در آینده‌ای نه چندان دور به سیستم‌های رمزی نیاز خواهیم داشت که در مقابل حملات مبتنی بر رایانه‌های کوانتومی مقاوم باشند که اصطلاحا این نوع سیستم‌های رمز را سیستم‌های رمز پساکوانتوم می‌نامند. رمزنگاری کدمبنا شاخه‌ای از رمزنگاری پساکوانتوم است که در آن امنیت رمزنگاری بر پایه مسائل سخت نظریه کدینگ کانال است [۱]. اولین سیستم رمز کدمبنا در سال ۱۹۷۸ توسط رابرت مک‌الیس ارائه شد، که در آن از کد گوپا باینری استفاده شده است. پس از آن طرح‌های مختلفی در این حوزه از جمله سیستم‌های رمزنگاری نامتقارن (کلید عمومی) کدمبنا، سیستم‌های رمز متقارن (کلید مخفی) کدمبنا ارائه شدند که هرکدام از طرح‌های ارائه شده مزایا و معایب مختلفی دارند. کارشناسان بر این عقیده‌اند که سیستم‌های رمز کدمبنا جزء سیستم‌های رمز پساکوانتوم می‌باشند و در برابر کامپیوترهای کوانتومی مقاومند و در آینده جایگزین خوبی برای سیستم‌های رمز کنونی که بر مبنای نظریه اعداد هستند، می‌باشند.

سیستم‌های رمزنگاری نامتقارن کدمبنا می‌توانند برای تبادل کلیدهای مخفی سیستم‌های رمزنگاری متقارن کدمبنا، که نسبتا کوتاه هستند، استفاده شوند. سیستم‌های رمزنگاری متقارن کدمبنا نیز می‌توانند برای رمزگذاری پیام‌های بلند استفاده شود. هر دو نوع سیستم‌های رمزنگاری متقارن و نامتقارن کدمبنا مزایا و معایب خود را دارند. الگوریتم‌های رمزنگاری متقارن کدمبنا سریعتر می‌باشند و به توان محاسباتی کمتری نیاز دارند، اما نقطه ضعف اصلی آنها توزیع کلید است. در مقابل، الگوریتم‌های رمزنگاری نامتقارن کدمبنا با استفاده از کلید عمومی برای رمزنگاری و بهره‌گیری از کلید خصوصی برای رمزگشایی، مشکل توزیع کلید را حل کرده‌اند. هر چند نقطه ضعف سیستم‌های رمزنگاری نامتقارن کدمبنا، طول بلندتر کلیدهای آن‌ها است. با استفاده از طرح رمزنگاری ترکیبی [i] (HES) کدمبنا می‌توانیم از مزایای هر دو سیستم‌های رمزنگاری متقارن و نامتقارن کدمبنا بهره‌مند شویم. یک طرح HES کدمبنا با ترکیب دو ساز و کار مجزا زیر قابل دستیابی است:

- ساز و کار لفافه‌بندی کلید [ii] (KEM) کدمبنا که برای تامین امنیت انتقال کلید مخفی سیستم رمزنگاری متقارن کدمبنا با استفاده از ساز وکارهای نامتقارن کدمبنا طراحی شده است. به طور کلی KEM از سه الگوریتم تشکیل شده است. الگوریتم نخست، الگوریتم تولید کلید Gen است که پارامتر امنیتی λ را به عنوان ورودی می‌گیرد و کلید عمومی لفافه‌گذاری (pk) و کلید خصوصی لفافه‌گشایی (sk) را به عنوان خروجی می‌دهد. الگوریتم دوم، الگوریتم لفافه‌گذاری کلید Encaps است که کلید عمومی لفافه‌گذاری pk را به عنوان ورودی می‌گیرد و در خروجی متن رمز شده $\bar{c}$ و کلید مخفی لفافه‌بندی داده $\mathcal{K} \in \{0,1\}^{\ell}$ را ارائه می‌دهد. در واقع، کلید مخفی $\mathcal{K}$ در متن رمز شده $(\bar{c})$ لفافه‌گذاری شده است. الگوریتم سوم، الگوریتم لفافه‌گشایی کلید Decaps که ورودی آن کلید خصوصی لفافه‌گشایی sk و متن رمزشده $\bar{c}$ است و خروجی آن کلید مخفی لفافه‌گشایی شده $\mathcal{K}$ یا شکست لفافه‌گشایی با نماد خاص $\perp$ است.
- ساز و کار لفافه‌بندی داده [iii] (DEM) کدمبنا که در واقع برای تامین امنیت انتقال پیام با استفاده از سازوکارهای رمزنگاری متقارن کدمبنا طراحی شده است. به طور کلی DEM نیز از سه الگوریتم تشکیل شده است. الگوریتم نخست، الگوریتم تولید کلید Gen است که پارامتر امنیتی λ را به عنوان ورودی می‌گیرد و کلید مخفی لفافه‌بندی داده $(\mathcal{K})$ را به عنوان خروجی می‌دهد. الگوریتم دوم، الگوریتم لفافه‌گذاری داده Encaps است که پیام m و کلید مخفی $\mathcal{K}$ را به عنوان ورودی می‌گیرد و متن لفافه‌گذاری شده (رمز شده) c را به عنوان خروجی می‌دهد. الگوریتم سوم، الگوریتم لفافه‌گشایی داده Decaps که ورودی آن کلید مخفی $\mathcal{K}$ و متن رمزشده c است و خروجی آن پیام لفافه‌گشایی شده m است. در واقع در طرح HES، زوج متن رمز شده $(c\ ,\ \bar{c})$ توسط فرستنده برای گیرنده ارسال می‌شود. گیرنده نیز پس از دریافت $(c\ ,\ \bar{c})$، عملیات لفافه‌گشایی کلید را بر $\bar{c}$ اعمال نموده و کلید مخفی لفافه‌گشایی شده $\mathcal{K}$ را بدست می‌آورد. سپس با استفاده از کلید لفافه‌گشایی شده $\mathcal{K}$ ، عملیات لفافه‌گشایی داده را بر c اعمال می‌نماید و پیام m را بازیابی می‌کند.

تاکنون چندین طرح رمزنگاری ترکیبی ارائه شده است که هرکدام مزایا و معایب خاص خود را دارند [۲-۳]. از طرفی کدهای قطبی [۴] دسته‌ای از کدهای قالبی خطی هستند که دارای پیچیدگی کم کدگذاری و کدگشایی می‌باشند. تاکنون از کدهای قطبی در طرح‌های رمزنگاری کلید عمومی، رمزنگاری کلید مخفی و رمزنگاری لایه فیزیکی استفاده شده است و قابلیت‌های مناسبی چه از نظر سطح امنیت و چه از نظر سطح کارآیی داشته‌اند [۵-۷]. در این مقاله از خصوصیات کدهای قطبی به منظور تامین امنیت و طراحی یک طرح HES پیشنهادی استفاده نموده‌ایم. سپس تحلیل‌های امنیت و کارآیی متداول را بر آن اعمال می‌نماییم. بخش‌های بعدی این مقاله به شرح زیر ارائه خواهند شد. در بخش ۲ با خواص کدهای قطبی و نحوه ساخت ماتریس مولد آنها آشنا می‌شویم. طرح HES پیشنهادی مبتنی بر کدهای قطبی در بخش ۳ بیان می‌شود که شامل دو بخش لفافه‌بندی کلید و لفافه‌بندی داده است. سطح امنیت و سطح کارآیی طرح HES پیشنهادی نیز در بخش ۳ بررسی می‌شوند. تحلیل امنیت طرح HES پیشنهادی با اعمال حمله جستجوی فراگیر کلید و حمله کدگشایی مجموعه اطلاعات و تحلیل کارآیی آن نیز بر حسب نرخ شکست رمزگشایی و طول کلید انجام شده است. در انتها خلاصه‌ای از نتایج در بخش ۴ ارائه شده است.

۲- کدهای قطبی

کدهای قطبی یک دسته از کدهای قالبی خطی هستند که به ظرفیت کانال‌های بدون حافظه گسسته با ورودی دودویی (B-DMC)[i] از قبیل کانال محک دودویی (BEC)[v] و کانال متقارن دودویی (BSC)[vi]، دست می‌یابند [۲]. کانال $W : \mathcal{X} \to \mathcal{Y}$ را به عنوان یک B-DMC با الفبای ورودی $\mathcal{X} = \{0,1\}$، الفبای خروجی $\mathcal{Y}$ و احتمالات انتقال $\{W(y|x), x \in \mathcal{X}, y \in \mathcal{Y}\}$ را در نظر می‌گیریم. پارامترهای $I(W)$ و $Z(W)$ را برای یک B-DMC W در نظر می‌گیریم که در آن $I(W) \in [0,1]$ اطلاعات متقابل بین ورودی و خروجی کانال انتقال W با توزیع یکنواخت بر ورودی است. علاوه بر این، $Z(W) \in [0,1]$ به عنوان پارامتر باتاکاریا کانال W شناخته می‌شود و به عنوان معیار اندازه‌گیری قابلیت اطمینان به کار می‌رود. برای هر B-DMC W، مجموعه‌ای $\left\{W_n^{(i)} : 1 \le i \le n\right\}$ از n کانال قطبی شده است که با اعمال پدیده‌ای به نام قطبش کانال بر n کپی مستقل از W به دست می‌آید. کانال‌های قطبی شده را اصطلاحاً بیت‌کانال‌ها یا زیرکانال‌ها می‌نامند. با استفاده از فرآیند قطبش کانال و با n به اندازه کافی بزرگ، ظرفیت $\left\{I\left(W_n^{(i)}\right), 1 \le i \le n\right\}$ و پارامتر باتاکاریا[vii] $\left\{Z\left(W_n^{(i)}\right), 1 \le i \le n\right\}$ هر کدام از بیت‌کانال‌ها به سمت صفر یا یک میل می‌کند. در این نوع کدها پس از اعمال قطبش کانال و تفکیک کانال‌ها به دو دسته کانال‌های خوب (بدون نویز) و کانال‌های بد (نویزی)، ورودی کانال‌های بد را ثابت (صفر) در نظر گرفته و اطلاعات را از کانال‌های خوب با قابلیت اطمینان بالا ارسال می‌کنیم.

۲-۱-ساخت ماتریس مولد کدهای قطبی

با توجه به اینکه در طرح HES پیشنهادی به ماتریس مولد کدهای قطبی نیاز داریم بنابراین در این بخش روش ساخت ماتریس مولد کدهای قطبی را نیز بررسی می‌کنیم. پارامترهای $n = 2^m$، $m \ge 1$ و $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ را در نظر می‌گیریم. با فرض $R < I(W)$ و $k = nR$، ماتریس مولد $G_{\mathcal{A}}$ برای کد قطبی (n,k) از طریق مراحل زیر ساخته می‌شود [۴]:

- ابتدا حاصلضرب کرونکر m ام ماتریس F که ماتریس $G_n = F^{\otimes m}$ به ابعاد $n \times n$ نتیجه می‌دهد را در نظر می‌گیریم و سطرهای ماتریس G_n را از بالا به پایین به صورت $i = 1, 2, \cdots, n$ شماره‌گذاری می‌کنیم.
- پارامترهای بلتاکاریا تمامی n بیت‌کانال‌ها $\left\{W_n^{(i)}, 1 \le i \le n\right\}$ از طریق فرمول بازگشتی (۱) با مقدار اولیه $Z_{1,1}$ به دست می‌آوریم و به صورت $Z_n = \left(Z_{n,i}, 1 \le i \le n\right)$ مرتب می‌کنیم.

$$Z_{2k,i} = \begin{cases} 2Z_{k,i} - Z_{k,i}^2 & 1 \le i \le k \\ Z_{k,i-k}^2 & k+1 \le i \le 2k \end{cases} \quad k = 1, 2, 2^2, \cdots, 2^{m-1} \qquad (۱)$$

- جایگشت $\pi_n = (i_1, i_2, \ldots, i_n)$ برای مجموعه اندیس‌های n بیت‌کانال، یعنی $\mathcal{I}_n = \{1, 2, \cdots, n\}$، را به دست می‌آوریم به نحوی که نامساوی $Z_{n,i_j} \le Z_{n,i_k}, 1 \le j < k \le n$ همواره برقرار باشد.
- مجموعه اطلاعات $\mathcal{A} \subset \mathcal{I}_n$ که اندیس بیت‌کانال‌های آن متناظر با k اندیس سمت چپ جایگشت π_n، یعنی $i_1, i_2, \ldots, i_k$ است را به دست می‌آوریم. سپس مجموعه ثابت $\mathcal{A}^c \subset \mathcal{I}$ که اندیس بیت‌کانال‌های آن متناظر با $n-k$ اندیس سمت راست جایگشت π_n، یعنی $i_{k+1}, i_{k+2}, \ldots, i_n$، است را به دست می‌آوریم.
- ماتریس مولد $G_{\mathcal{A}}$ را با انتخاب k سطر از ماتریس $G_n = F^{\otimes m}$، که اندیس‌هایشان متناظر با اندیس‌های بیت‌کانال‌های مجموعه اطلاعات $\mathcal{A}$ هستند، می‌سازیم. در واقع به ازای بیت‌کانال $W_n^{(i)}$، iامین ستون ماتریس G_n را انتخاب می‌کنیم. همچنین ماتریس $G_{\mathcal{A}^c}$ به ابعاد $(n-k) \times n$ را با انتخاب $n-k$ سطر ماتریس G_n، که اندیس‌هایشان متناظر با اندیس‌های بیت‌کانال‌های مجموعه ثابت $\mathcal{A}^c$ هستند را می‌سازیم.

۲-۲-کدگذاری کدهای قطبی

بردار ورودی $u = (u_1, u_2, \cdots, u_n) = (u_{\mathcal{A}}, u_{\mathcal{A}^c})$ شامل دو زیربردار است: الف) بردار اطلاعات $u_{\mathcal{A}} = (u_i, i \in \mathcal{A})$ که یک زیربردار k بیتی است؛ ب) بردارثابت $u_{\mathcal{A}^c} = (u_i, i \in \mathcal{A}^c)$ که یک زیربردار $n-k$ بیتی است. بردار اطلاعات $u_{\mathcal{A}}$ متشکل از بیت‌های اطلاعات است و می‌تواند در هر مرحله از انتقال تغییر کند در حالی‌که بردار ثابت متشکل از مقادیر ثابتی است که برای کدبردار حذف متوالی از قبل شناخته شده هستند. در کدهای قطبی غیرسیستماتیک بردار ورودی u به کلمه‌کد n بیتی x به صورت $x = u_{\mathcal{A}} G_{\mathcal{A}} + u_{\mathcal{A}^c} G_{\mathcal{A}^c} = u_{\mathcal{A}} G_{\mathcal{A}} + c$ کدگذاری می‌شود. از آنجا که $c \triangleq u_{\mathcal{A}^c} G_{\mathcal{A}^c}$ یک بردار ثابت است بنابراین کدگذاری به صورت غیرسیستماتیک است. در این حالت نرخ کد به صورت $R = |\mathcal{A}|/n$ تعریف می‌شود که می‌توان آن را با انتخاب اندازه مجموعه اطلاعات $\mathcal{A}$ تنظیم کرد.

۲-۳-کدگشایی کدهای قطبی

بردار x را به عنوان کلمه‌کد n بیتی که از n بیت‌کانال با اندیس‌های $\mathcal{I}_n = \{1, 2, \cdots, n\}$ عبور می‌کند را در نظر می‌گیریم. همچنین بردار y را به عنوان بردار خروجی متناظر که به وسیله الگوریتم حذف متوالی (SC)[viii] کدگشایی می‌شود را در نظر می‌گیریم. هدف کدبردار SC، تخمین بردار ورودی با آگاهی از مجموعه اطلاعات $\mathcal{A}$، بردار ثابت $u_{\mathcal{A}^c}$ و بردار خروجی کانال y به صورت رابطه (۲) است.

$$\hat{u}_i = \begin{cases} u_i, & i \in \mathcal{A}^c \text{ اگر} \\ h_i\left(y_1^n, \hat{u}_1^{i-1}\right) & i \in \mathcal{A} \text{ اگر} \end{cases} \qquad (۲)$$

که در آن توابع تصمیم‌گیری $h_i: \mathcal{Y}^n \times \mathcal{X}^{i-1} \to \mathcal{X}, i \in \mathcal{A}$ برای همه $y_1^n \in \mathcal{Y}^n, \hat{u}_1^{i-1} \in \mathcal{X}^{i-1}$ به صورت رابطه (۳) محاسبه می‌شوند،

$$h_i(y_1^n, \hat{u}_1^{i-1}) \triangleq \begin{cases} 0, & \text{اگر } \frac{W_n^{(i)}(y_1^n, \hat{u}_1^{i-1}|0)}{W_n^{(i)}(y_1^n, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{در غیر اینصورت} \end{cases} \quad (۳)$$

در واقع در تابع تصمیم‌گیری $\boldsymbol{h_i}$، بیت‌های اطلاعات $\boldsymbol{u_i}, \boldsymbol{i} \in \boldsymbol{\mathcal{A}}$ یک به یک و با استفاده از $\boldsymbol{i}$ امین عضو تصمیم‌گیری، بردار خروجی کانال $\boldsymbol{y}$ و بیت‌های اطلاعات تخمین زده شده قبلی $\boldsymbol{\hat{u}_1^{i-1}}$، تخمین زده می‌شوند. به علاوه مقادیر بیت‌های ثابت $\boldsymbol{u_i}, \boldsymbol{i} \in \boldsymbol{\mathcal{A}^c}$ برای کدبردار SC از قبل شناخته شده هستند.

۳-شرح سناریو و نتایج ارزیابی کارایی

طرح HES-PC پیشنهادی شامل دو بخش است: الف) لفافه‌بندی کلید پیشنهادی مبتنی بر کدهای قطبی (KEM-PC) که وظیفه آن لفافه‌گذاری، ارسال و لفافه‌گشایی کلید مخفی ($\mathcal{K}$) مورد نیاز در طرح لفافه‌بندی داده مبتنی بر کدهای قطبی است، ب) لفافه‌بندی داده پیشنهادی مبتنی بر کدهای قطبی (DEM-PC) که وظیفه آن لفافه‌گذاری (رمزگذاری)، ارسال و لفافه‌گشایی (رمزگشایی) داده (m) بین طرفین مجاز با استفاده از کلید مخفی $\mathcal{K}$ رد و بدل شده به وسیله طرح KEM-PC است. در این بخش، ساختار طرح‌های KEM-PC و DEM-PC پیشنهادی را بررسی می‌کنیم.

۳-۱-طرح لفافه‌بندی کلید پیشنهادی مبتنی بر کدهای قطبی (KEM-PC)

طرح KEM-PC پیشنهادی شامل سه بخش است: الف) الگوریتم تولید کلید (KEM-PC.Gen)، ب) الگوریتم لفافه‌گذاری کلید (KEM-PC.Encaps) و ج) الگوریتم لفافه‌گشایی کلید (KEM-PC.Decaps). در ادامه این زیربخش، ساختار سه الگوریتم یاد شده را بررسی می‌کنیم.

۳-۱-۱-الگوریتم تولید کلید طرح KEM-PC

ساختار الگوریتم KEM-PC.Gen به صورت زیر است:

- داده: سطح امنیتی λ.
- ورودی: بذر (s) مورد استفاده در مولد شبه تصادفی CTR-DRBG.
- خروجی: کلید عمومی لفافه‌گذاری (pk) $\mathcal{Q}$ و کلید خصوصی لفافه‌گشایی (sk) $\mathcal{A}^c(s)$.

۱. متناسب با λ داده شده، پارامترهای n، k، R و R_0 از کد قطبی (n, k) انتخاب می‌شوند.

۲. مجموعه اطلاعات مخفی $\mathcal{A}(s)$ به وسیله مولد شبه تصادفی CTR-DRBG ساخته می‌شوند که در واقع k اندیسی است که به طور تصادفی از n اندیس موجود در جایگشت π_n انتخاب شده است. سپس مجموعه $n-k$ عضوی $\mathcal{A}^c(s)$ به عنوان مکمل مجموعه $\mathcal{A}(s)$ به دست می‌آید و به عنوان کلید خصوصی لفافه‌گشایی sk ذخیره می‌شود.

۳. سپس ماتریس مولد مخفی $G_{\mathcal{A}(s)}$ برای کد قطبی (n, k) با انتخاب k سطر از G_n متناظر با $\mathcal{A}(s)$ تولید می‌شود.

۴. P_1 نیز به صورت یک زیرماتریس $n \times k$ که در آن تعداد k عدد یک متناظر با اندیس‌های $j \in \mathcal{A}(s)$ در ستون jام آن قرار داده شده است، تولید می‌شود. P_2 نیز به صورت یک زیرماتریس $n \times (n-k)$ که در آن تعداد $n-k$ عدد یک که هر کدام متناظر با اندیس‌های $j \in \mathcal{A}^c(s)$ در ستون jام آن قرار داده شده است، تولید می‌شود. در نهایت با کنار هم قرار دادن زیرماتریس‌های P_1 و P_2، ماتریس جایگشت به صورت $P = [P_1|P_2]$ تولید می‌شود.

۵. ماتریس غیرمنفرد نیز با انتخاب سطرها و ستون‌های متناظر با $\mathcal{A}(S)$ به صورت $S_{k\times k} = (G_n)_{\mathcal{A}(s)\mathcal{A}(S)}$ تولید می‌شود.

۶. ماتریس لفافه‌گذاری $G'_{k\times n}$ به صورت $S^{-1}G_{\mathcal{A}(S)}P = [I_k \,|\, Q]$ تولید می‌شود و زیرماتریس $Q_{k\times(n-k)}$ به عنوان کلید عمومی لفافه‌گذاری pk در نظر گرفته می‌شود.

۳-۱-۲-الگوریتم لفافه‌گذاری کلید طرح KEM-PC

ساختار الگوریتم لفافه‌گذاری KEM-PC.Encaps به صورت زیر است:

- ورودی: کلید عمومی pk.
- خروجی: کلید مخفی لفافه‌گذاری شده $\mathcal{K}$ و متن رمزشده c'.

۱. ماتریس G' به صورت $[I_k \,|\, Q]$ تولید می‌شود که در آن I_k یک ماتریس همانی به ابعاد $k \times k$ است.

۲. بردار خطای تصادفی $\bar{e} \epsilon F_2^n$ را با وزن همینگ $w_H(\bar{e}) \leq t$ و یک پیام تصادفی $\bar{m} \epsilon F_2^k$ تولید می‌شود.

۳. کلید لفافه‌گذاری شده $\mathcal{K}$ به صورت $\mathcal{K} = \mathrm{KDF}(\bar{m} \parallel \bar{e}, l_{\mathcal{K}})$ محاسبه می‌شود که در آن $\mathrm{KDF}: \{0,1\}^n \to \{0,1\}^{l_{\mathcal{K}}}$ یک تابع چکیده‌ساز با طول خروجی $l_{\mathcal{K}}$ است.

۴. متن رمز شده به صورت $\bar{c} = \bar{m}\bar{G} + \bar{e}$ محاسبه می‌شود.

۳-۱-۳-لفافه‌گشایی کلید طرح KEM-PC

ساختار الگوریتم لفافه‌گشایی کلید KEM-PC.Decaps به صورت زیر است:

- ورودی: متن رمزشده $\bar{c}$ و کلید خصوصی (sk) $\mathcal{A}^c(s)$.
- خروجی: کلید مخفی لفافه‌گشایی شده $\mathcal{K}$ یا نشانگر شکست $\perp$.

۱. ماتریس جایگشت P تولید می‌شود (شبیه به مرحله ۴ KEM-PC.Gen).

۲. سپس $\bar{c} = \bar{c}P^{-1} = \bar{m}S^{-1}G_{\mathcal{A}(S)} + eP^{-1}$ محاسبه می‌شود.
۳. مجموعه کلید خصوصی $\mathcal{A}^c(s)$ را مکمل نموده و مجموعه $\mathcal{A}(S)$ به دست آورده می‌شود.
۴. کدگشایی SC با ورودی $\{\bar{c}, \mathcal{A}(s)\}$ اعمال می‌شود و خروجی متناظر آن، یعنی $\bar{e} = \bar{e}P^{-1}$ و $\bar{u} = \bar{m}S^{-1}$، به دست می‌آید.
۵. اگر کدگشایی SC شکست بخورد یا $w_H(\bar{e}) \neq t$، خروجی $\perp$ است یا عملیات لفافه‌گشایی متوقف می‌شود.
۶. بردار $\bar{e} = \bar{\bar{e}}P$ به دست آورده می‌شود و ماتریس S تولید و $\bar{m} = \bar{u}S$ محاسبه می‌شود.
۷. کلید مخفی لفافه گشایی شده $\mathcal{K} = \text{KDF}(\bar{m} \parallel \bar{e}, l_{\mathcal{K}})$ محاسبه و بازیابی می‌شود.

۳-۲-طرح لفافه‌بندی داده پیشنهادی مبتنی بر کدهای قطبی (DEM-PC)

طرح DEM-PC پیشنهادی شامل سه الگوریتم است: الف) الگوریتم تولید کلید (DEM-PC.Gen)، ب) الگوریتم لفافه‌گذاری داده (DEM-PC.Encaps) و ج) الگوریتم لفافه‌گشایی داده (DEM-PC.Decaps). در ادامه این بخش، ساختار سه الگوریتم یاد شده را بررسی می‌کنیم.

۳-۲-۱-تولید کلید طرح DEM-PC

ساختار الگوریتم تولید کلید DEM-PC.Gen به صورت زیر است:

- داده: سطح امنیتی λ.
- ورودی: بذر (s) مورد استفاده در مولد شبه تصادفی CTR-DRBG.
- خروجی: کلید مخفی (sk) $\mathcal{A}^c(s)$.

۱. متناسب با λ داده شده، پارامترهای n، k و R از کد قطبی (n, k) انتخاب می‌شوند.
۲. مجموعه اطلاعات مخفی $\mathcal{A}(s)$ به وسیله مولد شبه تصادفی CTR-DRBG ساخته می‌شوند که در واقع k اندیسی است که به طور تصادفی از n اندیس موجود در جایگشت π_n است.
۳. سپس مجموعه $\mathcal{A}(s)$ به مجموعه $\mathcal{A}^c(s)$ مکمل می‌شود و به عنوان کلید خصوصی sk ذخیره می‌شود.

۳-۲-۲-لفافه‌گذاری داده طرح DEM-PC

ساختار الگوریتم لفافه‌گذاری داده DEM-PC.Encaps به صورت زیر است:

- ورودی: کلید مخفی (sk) $\mathcal{A}^c(s)$.
- خروجی: داده لفافه‌گذاری شده c.

۱. ابتدا مجموعه کلید مخفی $\mathcal{A}^c(s)$ به $\mathcal{A}(s)$ مکمل می‌شود و سپس ماتریس مولد مخفی $G_{\mathcal{A}(s)}$ برای یک کد قطبی (n, k) با انتخاب k سطر از G_n متناظر با $\mathcal{A}(s)$ تولید می‌شود.
۲. ماتریس جایگشت P تولید می‌شود (شبیه به مرحله ۴ KEM-PC.Gen).
۳. ماتریس غیرمنفرد S تولید می‌شود (شبیه به مرحله ۵ KEM-PC.Gen).
۴. متن لفافه‌گذاری شده (رمز شده) به صورت $c = (mSG_{\mathcal{A}(S)} + e)P$ محاسبه می‌شود که در آن $e \epsilon F_2^n$ بردار خطای تصادفی با وزن همینگ $w_H(e) \leq t$ و $m \epsilon F_2^k$ بردار پیام است.

۳-۲-۳-لفافه‌گشایی داده طرح DEM-PC

ساختار الگوریتم لفافه‌گشایی داده DEM-PC.Decaps به صورت زیر است:

- ورودی: متن رمز شده c و کلید خصوصی $\mathcal{A}^c(s)$.
- خروجی: کلید لفافه‌گشایی $\mathcal{K}$ یا نشانگر شکست $\perp$.

۱. ابتدا مجموعه کلید مخفی $\mathcal{A}^c(s)$ به مجموعه $\mathcal{A}(s)$ مکمل می‌شود.
۲. ماتریس جایگشت P تولید می‌شود (شبیه به مرحله ۴ KEM-PC.Gen).
۳. سپس $c' = cP^{-1} = mG_{\mathcal{A}(S)} + e$ محاسبه می‌شود.
۴. کدگشایی حذف متوالی (SC) با ورودی $\{c', \mathcal{A}(s)\}$ اعمال می‌شود و بردار $u = mS$ به عنوان خروجی آن بدست می‌آید.
۵. اگر کدگشایی SC شکست بخورد یا $w_H(e') \neq t$، خروجی $\perp$ است یا عملیات لفافه‌گشایی داده متوقف می‌شود.
۶. ماتریس غیرمنفرد S تولید می‌شود (شبیه مرحله ۵ KEM-PC.Gen).
۷. بردار $m = uS^{-1}$ محاسبه می‌شود.

۳-۳-تحلیل امنیت طرح HES-PC

در این بخش تحلیل امنیتی طرح پیشنهادی HES-PC با اعمال حمله جستجوی فراگیر کلید و حمله کدگشایی مجموعه اطلاعات انجام می‌شود.

۳-۳-۱-حمله جستجوی فراگیر کلید

حمله جستجوی فراگیر کلید نوعی حمله ساختاری است که در آن تمام کلیدهای محتمل جستجو می‌شود تا کلید مناسب شناسایی شود. هرچند این حمله اگر فضای کلید خصوصی به اندازه کافی بزرگ باشد، شکست می‌خورد. در طرح HES-PC پیشنهادی به دلیل انتخاب تصادفی k زیرکانال خوب از n زیرکانال، تعداد مجموعه کلید مخفی $\mathcal{A}^c(s)$ برابر است با $\mathcal{N}_{\mathcal{A}^c(s)} = \binom{n}{k}$. به طور مثال، برای کد قطبی $(256, 192)$، $\mathcal{N}_{\mathcal{A}^c(s)}$ در حدود $2^{203.57}$ خواهد بود. تعداد ماتریس‌های غیرمنفرد برابر است با تعداد تمام زیرماتریس‌های

$G_{i,j}$ و G_n با اندیس‌های $i,j \in \mathcal{A}(s)$، این یعنی $\mathcal{N}_S = \binom{n}{k}$. همچنین تعداد ماتریس‌های جایگشت دودویی $P_{n\times n}$ به صورت $\mathcal{N}_P = \mathcal{N}_{P_1}.\mathcal{N}_{P_2} = \binom{n}{k} \times (n-k)$ محاسبه می‌شود. جدول (۱) فاکتور کار (میزان عملیات باینری مورد نیاز) شمارش $\mathcal{A}^c(s)$، S و P برای نمونه‌های HES-PC را نشان می‌دهد.

جدول ۱- فاکتور کار (WF) حمله جستجوی فراگیر کلید برای نمونه های طرح HES-PC پیشنهادی

نمونه‌ها	(n,k)	فاکتور کار (log_2) شمارش S و $\mathcal{A}^c(s)$	فاکتور کار (log_2) شمارش P
HES-PC I	$(256,192)$	203.57	209.57
HES-PC II	$(512,384)$	410.76	417.76
HES-PC III	$(1024,768)$	825.63	833.63
HES-PC IV	$(2048,1536)$	1655.88	1664.88

۳-۳-۲-حمله کدگشایی مجموعه اطلاعات

الگوریتم کدگشایی مجموعه اطلاعات (ISD) یکی از قوی‌ترین الگوریتم‌هایی است که برای حل مسئله کدگشایی سیندروم به کار می‌رود و غالبا پیچیدگی اجرای این حمله است که سطح امنیتی اغلب سیستم‌های رمزنگاری کدمبنا را مشخص می‌کند. مجموعه اطلاعاتی کد C، که با مجموعه I نشان داده می‌شود، نمایانگر k جایگاه در متن رمز شده است که ستون‌های متناظر آن در H یعنی H_l تشکیل یک ماتریس وارون‌پذیر را بدهد. هر الگوریتم ISD امیدوار است که که بردار خطا نسبت به مجموعه اطلاعاتی داده شده، دارای یک مدل خطای معین باشد. اولین بار این حمله توسط McEliece مطرح شد. پس از او در کارهای متعددی بهبودهایی در این حمله داده شد. یکی از قوی‌ترین انواع الگوریتم‌های ISD توسط Stern مطرح شد [۸].

پس از Stern نیز کارهای متعدد دیگری در این حوزه انجام شده است، اما از آن جایی که غالبا محاسباتشان به صورت مجانبی بوده است و نمی‌توان پیچیدگی آن‌ها را به صورت دقیق محاسبه نمود، ما در این مقاله از الگوریتم Stern برای محاسبه پیچیدگی استفاده می‌کنیم [۸]. Stern چینش خطا را به صورت دو قسمت p تایی با فرض l ستون تمام صفر در نظر گرفت و با استفاده از تناقض نما روز تولد پیچیدگی را بدست آورد. احتمال موفقیت الگوریتم Stern در یافتن یک کلمه‌کد با وزن همینگ داده شده w و با شرایط ذکر شده، به صورت رابطه (۴) محاسبه می‌شود که در آن $0 \le p \le \omega$ و $0 \le \ell \le n-k$ دو عدد صحیح به عنوان پارامترهای الگوریتم است و مقدار آنها به گونه‌ای تعیین می‌شود که پیچیدگی حمله حداقل گردد.

$$P_{ST} = \binom{k/2}{p}^2 \binom{n-k-\ell}{\omega-2p} \binom{n}{\omega}^{-1} \qquad (۴)$$

پیچیدگی هر بار اجرای الگوریتم نیز به صورت رابطه (۵) محاسبه می‌شود.

$$Cost_{ST} = \frac{1}{2}(n-k)^2(n+k) + 2\binom{k/2}{p} p\ell + 2p(n-k)\binom{k/2}{p}^2 / 2^{\ell} \qquad (۵)$$

در نهایت فاکتور کار حمله ISD با رابطه (۶) محاسبه می‌شود.

$$\mathrm{WF}_{\mathrm{isd}}(n,k,\omega) = Cost_{ST}/P_{ST} \qquad (۶)$$

جدول (۲) فاکتور کار نمونه‌های مختلف طرح HES-PC پیشنهادی در برابر حمله ISD به ازای طول‌های مختلف کد قطبی و پارامتر را با استفاده از الگوریتم Stern برای سه دسته پارامتر نشان می دهد. همان‌طور که در این جدول مشاهده می‌شود نمونه‌های HES-PC III و HES-PC IV مقاومت مناسبی در برابر حمله ISD دارند.

جدول ۲-فاکتور کار (WF) حمله ISD برای نمونه‌های مختلف طرح HES-PC پیشنهادی

نمونه‌ها	(n,k)	(p,ℓ)	$(log_2)\mathrm{WF}_{\mathrm{ISD}}$
HES-PC I	$(256,192)$	(2,8)	79.96
HES-PC II	$(512,384)$	(3,22)	104.61
HES-PC III	$(1024,768)$	(5,39)	140.63
HES-PC IV	$(2048,1536)$	(7,59)	190.19

٤-تحلیل کارآیی طرح HES-PC

در این بخش، کارآیی طرح HES-PC پیشنهادی را بر حسب نرخ شکست رمزگشایی و طول کلید مورد ارزیابی قرار می‌دهیم.

٤-١-نرخ شکست رمزگشایی

در این زیربخش، نرخ شکست رمزگشایی (DFR) طرح HES-PC پیشنهادی را تحت کدگشایی SC تحلیل می‌کنیم. در طرح HES-PC پیشنهادی، کدگشایی SC مقدار i امین بیت ورودی که با $\hat{u}_i$ نشان داده می‌شود را توسط بردار دریافت شده $y = y_1^n$ و بیت‌های ورودی تخمین زده شده قبلی $\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_{i-1}$، تخمین می‌زند. در واقع، DFR احتمال شکست کدگشای حذف متوالی است، وقتی که مجموعه اطلاعات مخفی $\mathcal{A}(s)$ به صورت تصادفی از n زیرکانال انتخاب شده باشند. در ادامه، تحلیل بیشترین و کمترین کران بالایDFR در طرح HES-PC پیشنهادی تحت کدگشایی حذف متوالی بیان می‌شود.

فرض کنید$\mathcal{A}_1 = \{i \in \mathcal{I}_n : \pi_n(i) \in \{i_1, i_2, \cdots, i_k\}\}$ یک زیرمجموعه از $\mathcal{A}$ است که لندیس‌های آن در مجموعه $\mathcal{I}_n$ مربوط به k اندیس $i_1, i_2, \cdots, i_k$ از جایگشت π_n است. کمترین کران بالای DFR تحت کدگشایی SC، DFR_{min}، مجموع پارامترهای باتاکاریا k بیت کانال است با اندیس‌هایی از زیرمجموعه $\mathcal{A}$ است یعنی $\mathrm{DFR}_{min} = \sum_{i\in\mathcal{A}_1} Z(W_n^{(i)})$. همچنین فرض کنید $\mathcal{A}_2 = \left\{i \in \mathcal{I} : \pi_n(i) \in \{i_1, i_2, \cdots, i_{nR_0}\}\right\}$ زیرمجموعه‌ای از $\mathcal{A}$ است که اندیس‌های آن در مجموعه $\mathcal{I}_n$ مربوط به k اندیس $i_{n+1-k}, i_{n+2-k}, \cdots, i_{nR_0}$ از جایگشت π_n است. بیشترین کران بالای DFR تحت کدگشایی SC، DFR_{max}، مجموع

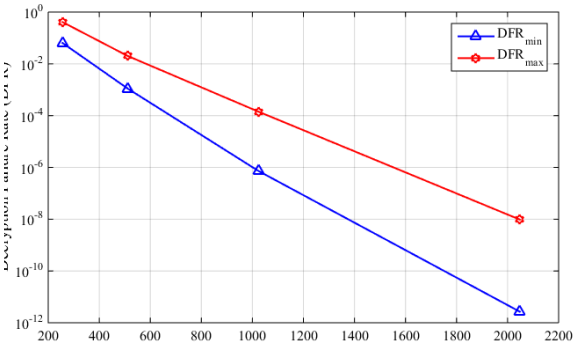
پارامترهای باتاکاریا k بیت کانال با اندیس‌های زیرمجموعه $\mathcal{A}$ است یعنی داریم $\mathrm{DFR}_{max} = \sum_{i\in\mathcal{A}_2} Z(W_n^{(i)})$.

بنابراین، در طرح HES-PC پیشـــنهادی کران بالای DFR در کدگشـایی SC، متناسـب با مقدار پارامتر باتاکاریای k بیت کانالی که به طور تصـادفی انتخاب شـده‌اند، می‌تواند از DFR_{min} تا DFR_{max} تغییر می‌کند. جدول (۳) کمترین و بیشـــترین کران بالای DFR را برای HES-PC پیشنهادی نشان می‌دهد.

جدول ۳-کمترین و بیشترین کران بالای DFR برای نمونه‌های مختلف طرح HES-PC پیشنهادی

نمونه‌ها	DFR_{min}	DFR_{max}
HES-PC I	64×10^{-3}	41×10^{-2}
HES-PC II	11×10^{-4}	2×10^{-2}
HES-PC III	7.26×10^{-7}	1.42×10^{-4}
HES-PC IV	2.67×10^{-12}	9.64×10^{-9}

شـــکل (۱) نیز مقادیر DFR_{min} و DFR_{max} را برای نمونه‌های مختلف طرح HES-PC پیشـنهادی را نشـان می‌دهد. همانطور که در شـکل مشـاهده می‌شـود، DFR_{min} و DFR_{max} با افزایش طول کد کاهش می‌یابد. اختلاف بین DFR_{min} و DFR_{max} در HES-PC IV بیش از سایر نمونه‌ها است.



شکل ۱- DFR_{min} و DFR_{max} برای نمونه‌های مختلف HES-PC

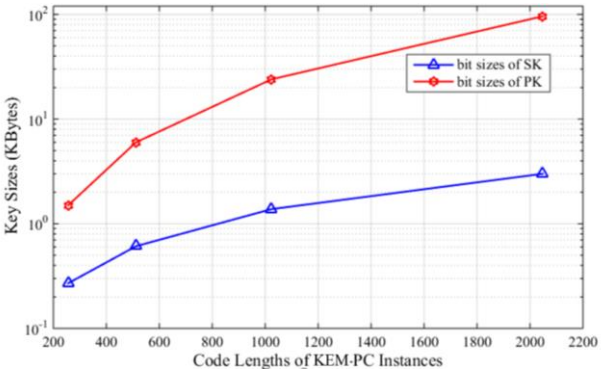
٤-۲-اندازه کلید

در این بخش طول کلیدهای pk و sk برای نمونه‌های مختلف HES-PC محاسبه می‌شود. در امنیت CCA2 ماتریس رمزگذاری را می‌توان سیستماتیک درنظر گرفت که به جای kn بیت، $k(n-k)$ بیت را اشغال می‌کند [۹]. به دلیل اینکه HES-PC دارای امنیت CCA2 است می‌توانیم از ماتریس رمزگذاری سیستماتیک $G' = [I_k|Q]$ استفاده کنیم. اندازه کلید عمومی $Q_{k\times(n-k)}$ برابر با $k(n-k)$ بیت است. کلید خصوصی در HES-PC شامل مجموعه $\mathcal{A}^c(s)$ است. هر اندیس $\mathcal{A}^c(s)$ برای ذخیره‌سازی نیاز به $log_2(n)$ بیت دارد [۱۰]. کران بالای حافظه برای ذخیره‌سازی کلید خصوصی $\mathcal{A}^c(s)$، $(n-k)log_2(n)$ بیت است. جدول (۴) اندازه طول کلید pk و sk را برای HES-PC پیشنهادی نشان می‌دهد. همانگونه که نشان داده شده، HES-PC IV بزرگترین طول کلیدهای خصوصی و عمومی دارد.

جدول ٤- حداکثر طول کلید خصوصی و عمومی در نمونه‌های مختلف طرح HES-PC

نمونه ها	(kBytes)sk	(kBytes)pk
HES-PC I	0.06	1.5
HES-PC II	0.14	6
HES-PC III	0.31	24
HES-PC IV	0.69	96

همچنین شکل (۲) طول کلید عمومی (pk) و کلید خصوصی (sk) برای طول کدهای مختلف در طرح HES-PC پیشنهادی نشان می‌دهد. همانطور که در این شکل نشان داده شده است، طول کلید عمومی و خصوصی با افزایش طول کد بیشتر می‌شود. همچنین طول کلید خصوصی همواره از طول کلید عمومی برای طول‌های مختلف کد قطبی در طرح HES-PC پیشنهادی کمتر است.



شکل ۲- طول کلید PK و SK برای طول کدهای مختلف در KEM-PC

۵-نتیجه گیری

در این مقاله یک طرح رمزنگاری ترکیبی کدمبنا ارائه شده است که در آن لفافه‌بندی کلید و لفافه‌بندی داده بر اساس خواص کدهای قطبی با یکدیگر ترکیب شده‌اند. امنیت طرح پیشنهادی بر اساس سختی مسئله کدگشایی عام است. با بهره‌گیری از خواص کدهای قطبی، طول کلید عمومی و خصوصی برای نمونه‌های مختلف طرح رمزنگاری ترکیبی پیشنهادی، کوتاه و کاربردی شده است. همچنین نرخ شکست رمزگشایی با افزایش طول کد نیز کاهش می‌یابد و بیشترین و کمترین کران بالای آن برای نمونه‌های با طول کد بیشتر، دارای مقادیر بهینه هستند. فاکتور کار اعمال حمله کدگشایی مجموعه اطلاعات بر طرح رمزنگاری ترکیبی پیشنهادی نیز به ازای طول کد $n = 1024$، برابر با $2^{140.63}$ است که نمایانگر سطح امنیت مناسب طرح پیشنهادی در برابر این حمله است. علاوه بر آن، فاکتور کار طرح پیشنهادی در برابر حمله جستجوی فراگیر کلید برای نمونه‌های مختلف دارای مقادیر بیش از 2^{200} است که نشان می‌دهد فضای کلید به اندازه کافی بزرگ است.

مراجع

[1] McEliece, R. J., "*A public-key cryptosystem based on algebraic coding theory*," DNS Progress Report, Jet Propulsion Laboratory, CA, Pasadena, pp. 114-116, 1978.

[2] Perumal, P. K., "*Hybrid Encryption/Decryption Technique Using New Public Key and Symmetric Key Algorithm*,"

International Journal of Information and Computer Security, vol. 6, p. 372, 2014.

[3] Singha, R., "*Hybrid Encryption Scheme (HES): An Approach for Transmitting Secure Data over Internet*," Procedia Computer Science, vol. 48, pp. 51-57, 2015.

[4] Arıkan, E., "*Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*," IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3051-3073, July 2009.

[5] Hooshmand, R., Aref, M. R., "*Polar Code-based Secure Channel Coding Scheme with Small Key Size*," IET Commun., vol. 11, no. 15, pp. 2357 - 2361, 2017.

[6] Hooshmand, R., Koochak Shooshtari, M., and Aref, M. R., "*PKC-PC: A Variant of the McEliece Public Key Cryptosystem based on Polar Codes*", IET Commun., Vol. 14, no. 12, pp. 1883 - 1893, 2020.

[7] Hooshmand, R., Aref, M. R., Eghlidos, T., "*Physical layer encryption scheme using finite-length polar codes*," IET Commun., vol. 9, no. 15, pp. 1857-1866, 2015.

[8] Stern, J., "*A method for finding codewords of small weight*," in Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings, pp.106–113, 1988.

[9] R. Hooshmand, A. Jafari and G. Karamali, "*Id-PC: An Identification Scheme based on Polar Codes*," Information Security Journal: A Global Perspective pp. 1-15, 2022.

[10] R. Hooshmand and M. Khoshfekr, "*Key encapsulation mechanism based on polar codes*," IET Communications, pp. 2438-2447, 2022.