دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۰، جلد ۱۹، شماره ۲ (۲۲-۳۱)
https://monadi.isc.org.ir

ارائه‌شده در هفدهمین کنفرانس بین المللی انجمن رمز ایران در دانشگاه علم و صنعت ایران، ۱۹ و ۲۰ شهریور ۱۳۹۹

ارائه یک طرح شناسایی امن و کارآ مبتنی بر کدهای قطبی★

علیرضا جعفری*[۱،۲]، رضا هوشمند[۱]، معصومه کوچک شوشتری[۲] و غلامرضا کرمعلی[۳]

[۱]مرکز تحصیلات تکمیلی، دانشگاه هوایی شهید ستاری، تهران، ایران
[۲]دانشکده مهندسی برق، دانشگاه صنعتی شریف، تهران، ایران
[۳]دانشکده علوم پایه، دانشگاه هوایی شهید ستاری، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
رمزنگاری پساکوانتوم
طرح شناسایی Stern
کدهای قطبی

dor: 10.0000/000000000

نوع مقاله: پژوهشی

چکیده

رمزنگاری کدمبنا یکی از نامزدهای رمزنگاری پساکوانتوم است که در سال‌های اخیر تحقیقات فراوانی در این حوزه انجام شده است. یکی از شاخه‌های رمزنگاری کدمبنا، طرح‌های شناسایی کدمبنا هستند که می‌توان از آنها جهت احراز هویت بین طرفین استفاده نمود. در این مقاله، با استفاده از طرح شناسایی Stern، یک طرح شناسایی کدمبنا ارائه می‌کنیم که در آن به جای استفاده از کدهای تصادفی از کدهای قطبی استفاده شده است. یکی از خواص کدهای قطبی این است که می‌توان ماتریس مولد و ماتریس توازن‌آزما آنها را با توجه به خصوصیات کانال از روی ماتریس آشکار بدست آورد. این خاصیت به ما کمک می‌کند که در طرح شناسایی کدمبنا پیشنهادی مجبور نباشیم کل ماتریس مولد یا توازن‌آزما را به عنوان داده عمومی ذخیره نماییم. این راهکار به طور قابل ملاحظه‌ای باعث کاهش حافظه مورد نیاز جهت ذخیره داده عمومی می‌شود. همچنین در طرح شناسایی پیشنهادی با استفاده از راهکارهایی هزینه ارتباطات نیز نسبت به طرح شناسایی Stern کاهش می‌یابد. سطح امنیت طرح شناسایی پیشنهادی نیز از دو دیدگاه احتمال تقلب و اثبات هیچ‌آگاهی مورد بررسی قرار گرفته است. علاوه بر آن، نشان می‌دهیم که طرح شناسایی پیشنهادی در برابر حمله کدبرداری مجموعه اطلاعات مقاوم است.

© ۱۴۰۰ انجمن رمز ایران

۱ مقدمه

با ظهور کامپیوترهای کوانتومی با توان محاسباتی بالا در آینده‌ای نزدیک، سیستم‌های رمزنگاری کلید عمومی متداول کنونی مانند RSA، ElGamal و ECC با خطر شکسته شدن مواجه می‌شوند. بنابراین 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 کدنگاری کانال است. در واقع سیستم‌های رمز کدمبنا در صورت طراحی مناسب می‌توانند در برابر حملات مبتنی بر رایانه‌های کوانتومی مقاومت مناسبی داشته باشند و در آینده جایگزین مناسبی برای سیستم‌های رمز کلید عمومی کنونی که بر مبنای نظریه اعداد هستند، باشند. یکی دیگر از برتری‌های استفاده از سیستم‌های رمز کدمبنا نسبت به طرح‌هایی که امروزه به صورت گسترده

★از کمیته علمی هفدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول
آدرس‌های رایانامه: jafarialireza59@gmail.com (علیرضا جعفری)، rhooshmand@ssau.ac.ir (رضا هوشمند)، m.koochak@ee.sharif.edu (معصومه کوچک شوشتری)، gh_karamali@azad.ac.ir (غلامرضا کرمعلی)
© ۱۴۰۰ تمامی حقوق متعلق به انجمن رمز ایران است.

استفاده می‌شوند، سرعت بالای رمزنگاری و رمزگشایی آنهاست.

نخستین سیستم رمز کلید عمومی کدمبنا در سال ۱۹۷۸ توسط McEliece ارائه شد، که در آن از کد گوپا باینری استفاده شده است [۱]. این سیستم رمز از بسیاری از سیستم‌های رمز کلید عمومی متداول از جمله RSA یا طرح‌های رمز مبتنی بر خم بیضوی سریعتر است. پس از آن طرح‌های مختلفی در این حوزه از جمله سیستم‌های رمز کلید عمومی کدمبنا، سیستم‌های رمز کلید مخفی کدمبنا، امضاهای دیجیتال کدمبنا و طرح‌های شناسایی کدمبنا ارائه شدند که هر کدام از طرح‌های ارائه شده مزایا و معایب مختلفی دارند.

امروزه با توجه به تحولات زیاد در سیستم‌های مخابراتی و شبکه‌های اینترنتی، یکی از مسائل مهم در حوزه امنیت این نوع سیستم‌ها و شبکه‌ها، بحث احراز هویت بین طرفین مجاز است. یکی از شاخه‌های رمزنگاری کدمبنا، طرح‌های شناسایی کدمبنا هستند که می‌توان از آنها جهت احراز هویت بین طرفین استفاده نمود. نخستین طرح شناسایی کدمبنا در سال ۱۹۹۳ توسط Stern ارائه شده است [۲]. این طرح شناسایی هیچ‌آگاهی، یک پروتکل سه مرحله‌ای است که در آن هر دور در سه مرحله انجام می‌شود. امنیت این طرح شناسایی متکی به مسئله سخت کدبرداری سیندروم است که ثابت شده است یک مسئله NP-Complete می‌باشد [۳]. در این طرح احتمال تقلب در هر دور اجرای پروتکل برابر با $\frac{2}{3}$ است. بنابراین برای احراز هویت باید پروتکل چندین بار تکرار شود و اگر پروتکل را r بار تکرار کنیم، احتمال تقلب برابر با $(\frac{2}{3})^r$ می‌شود. سپس طرح شناسایی Stern توسط Veron به یک پروتکل شناسایی معادل تبدیل شد که امنیت آن بر مبنای مسئله کدبرداری عام است [۴].

طرح‌های شناسایی Stern و Veron دو ضعف اصلی دارند، اول اینکه داده عمومی این طرح‌ها که شامل ماتریس توازن‌آزما یا ماتریس مولد کد مورد استفاده است، فضای زیادی برای ذخیره نیاز دارد. ضعف دوم این است که به دلیل احتمال تقلب، این پروتکل‌های شناسایی بایستی بارها تکرار شوند و در نتیجه هزینه ارتباطات در این طرح‌ها نیز زیاد است.

تاکنون طرح‌هایی بر مبنای طرح شناسایی Stern ارائه شده‌اند که در هر یک از آنها کارآیی این طرح تا حدی بهبود یافته است [۵، ۶]. در سال ۲۰۰۷ طرحی مشابه طرح Stern ارائه شد که در آن از کدهای دوری استفاده شد و اندازه داده‌های عمومی را به طور قابل ملاحظه‌ای کاهش داد [۶]. در این طرح از ماتریس‌های دوری استفاده شده است که حجم حافظه مورد نیاز جهت ذخیره داده عمومی را تا چند صد بیت کاهش داد. در یک ماتریس دوری با استفاده از سطر اول و چرخش آن می‌توان کل ماتریس را تولید کرد. در سال ۲۰۱۱ نسخه دیگری از طرح Veron ارائه شده است که شامل یک پروتکل ۵ مرحله‌ای است و در آن از کدهای دوری استفاده می‌شود [۷]. در این طرح نیز احتمال تقلب تقریبا برابر $\frac{1}{2}$ می‌باشد، که باعث می‌شود تعداد تکرار کمتر شود و در نتیجه هزینه ارتباطات کلی نیز کم شود. همچنین در این نسخه از طرح شناسایی راهکارهایی برای فشرده‌سازی و کم کردن حجم داده‌های ارسالی ارائه شده است که به طور کلی هزینه ارتباطات را کاهش می‌دهد.

یکی از راهکارهای موجود جهت کاهش اندازه داده عمومی طرح‌های شناسایی Stern و Veron استفاده از کدهای قطبی در ساختار آنها است. در این مقاله بر مبنای طرح شناسایی Stern، یک طرح شناسایی مبتنی بر کدهای قطبی ارائه می‌کنیم که در آن به جای استفاده از کدهای تصادفی از کدهای قطبی استفاده شده است. یکی از خواص کدهای قطبی این است که می‌توان ماتریس مولد و ماتریس توازن‌آزما آنها را از روی ماتریس آشکار G_n بدست آورد. این خاصیت به ما کمک می‌کند که در طرح‌های رمز کدمبنا مجبور نباشیم کل ماتریس را ذخیره نماییم. در ادامه این مقاله از این خاصیت کدهای قطبی در جهت ارتقاء امنیت و کارآیی طرح شناسایی Stern بهره می‌گیریم. استفاده از کدهای قطبی در ساختار این طرح‌ها باعث کم شدن حجم داده‌های عمومی آن می‌شود، به طوریکه بجای یک ماتریس $k \times n$ فقط یک بردار nبیتی را ذخیره می‌نماییم. همچنین برای کاهش هزینه ارتباطات از روش کاهش تعداد چکیده‌های ارسالی در هر دور اجرای پروتکل که در [۶] به آن اشاره شده است نیز استفاده می‌کنیم. همچنین می‌توان از روش تولید بذر و ارسال آن بجای مقدار تصادفی [۸] و فشرده سازی کلمات با وزن کم [۷] نیز جهت کم کردن هزینه ارتباطات استفاده کرد. علاوه بر این، امنیت این طرح شناسایی پیشنهادی را از دیدگاه احتمال تقلب و اثبات هیچ‌آگاهی و حمله کدبرداری مجموعه اطلاعات بررسی می‌کنیم.

بخش‌های بعدی این مقاله به شرح زیر ارائه خواهند شد. در بخش ۲ با ساختار طرح شناسایی Stern و نقاط ضعف آن آشنا می‌شویم. خواص کدهای قطبی و نحوه ساخت ماتریس توازن‌آزما آنها نیز در بخش ۳ ارائه می‌شود. با توجه به نقاط ضعف طرح شناسایی Stern، طرح شناسایی پیشنهادی مبتنی بر کدهای قطبی در بخش ۴ بیان می‌شود. سطح امنیت و سطح کارآیی طرح پیشنهادی به ترتیب در بخش‌های ۵ و ۶ بررسی می‌شوند. در انتها خلاصه‌ای از نتایج در بخش ۷ ارائه شده است.

۲ طرح شناسایی Stern

نخستین طرح شناسایی کدمبنا مبتنی بر سختی کدبرداری سیندروم است و در سال ۱۹۹۳ توسط Stern [۲] ارائه شده است. این طرح دارای یک پروتکل سه مرحله‌ای می‌باشد که اثبات‌کننده [1] و وارسی‌کننده [2] باید بارها آن را تکرار نمایند تا به سطح امنیتی مورد نیاز برسند. در ادامه این بخش ساختار طرح شناسایی Stern را توضیح می‌دهیم.

داده های عمومی: پارامترهای $n, k, w \in \mathbb{N}$ و یک ماتریس توازن‌آزما H به ابعاد $(n-k) \times n$ بر روی $\mathbb{F}_2$ و تابع چکیده ساز h

کلید خصوصی: $s \in W_{2,n,w}$ کلید nبیتی با وزن همینگ w

کلید عمومی: $S = Hs^T$

تمهیدات: اثبات‌کننده به صورت تصادفی $y \in \mathbb{F}_2^n$ و جایگشت π از $\{1, \ldots, n\}$ را انتخاب می‌کند و سپس $c_1 = h(\pi, Hy^T)$ و $c_2 =$

[1] prover [2] verifier

$h(\pi(y))$ و $c_3 = h(\pi(y+s))$ را حساب می‌کند و c_1، c_2 و c_3 را برای وارسی‌کننده می‌فرستد.

چالش: وارسی‌کننده مقدار $b \in \{0,1,2\}$ را به صورت تصادفی انتخاب می‌نماید و برای اثبات‌کننده می‌فرستد.

پاسخ: سه حالت داریم که در هر حالت اثبات‌کننده، z را به عنوان پاسخ بر می‌گرداند. اگر $b = 0$ بود آنگاه $z = (y, \pi)$. اگر $b = 1$ بود آنگاه $z = (y+s, \pi)$. اگر $b = 2$ بود آنگاه $z = (\pi(y), \pi(s))$.

تایید: سه حالت وجود دارد که وارسی‌کننده پس از بررسی صحت حالت‌های مختلف پروتکل را تایید می نماید. اگر $b = 0$ بود، وارسی‌کننده صحت c_1 و c_2 را بررسی می‌کند. اگر $b = 1$ بود، وارسی‌کننده صحت c_1 و c_3 را بررسی می کند. اگر $b = 2$ بود، وارسی‌کننده صحت c_2 و c_3 را بررسی می‌کند.

با توجه به اینکه احتمال تقلب پروتکل دقیقا برابر $\frac{2}{3}$ است، یک اثبات‌کننده حقیقی برای پذیرش، باید بارها و بارها پروتکل را تکرار کند. در مقاله [۲] ۳۵ تکرار پیشنهاد شده است که منجر به احتمال تقلب $2^{-20} \approx 10^{-6}$ می‌شود و به نظر می‌رسد سطح احراز هویت ضعیفی باشد.

۳ کدهای قطبی

کدهای قطبی یک دسته از کدهای قالبی خطی هستند که به ظرفیت کانال‌های بدون حافظه گسسته با ورودی دودویی (B-DMC [1]) از قبیل کانال محک دودویی (BEC [2]) و کانال متقارن دودویی (BSC [3])، دست می‌یابند [۹]. کانال $W : X \to Y$ را به عنوان یک B-DMC با الفبای ورودی $X = \{0,1\}$، الفبای خروجی Y و احتمالات انتقال $\{W(y|x), x \in X, y \in Y\}$ 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، $\{W_n^{(i)} : 1 \le i \le n\}$ مجموعه‌ای از n کانال قطبی شده است که با اعمال پدیده‌ای به نام قطبش کانال [4] بر n کپی مستقل از W به دست می‌آید. کانال‌های قطبی شده را اصطلاحاً بیت‌کانال‌ها [5] یا زیرکانال‌ها می‌نامند. با استفاده از فرآیند قطبش کانال و با n به اندازه کافی بزرگ، ظرفیت $\{I(W_n^{(i)}), 1 \le i \le n\}$ و پارامتر باتاکاریا [6] $\{Z(W_n^{(i)}), 1 \le i \le n\}$ 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. با توجه به اینکه در طرح شناسایی پیشنهادی به ماتریس توازن‌آزما کدهای قطبی نیاز داریم بنابراین در این بخش روش ساخت ماتریس توازن‌آزما کدهای قطبی را نیز بررسی می‌کنیم.

مجموعه $\mathcal{I}_n = \{i, 1 \le i \le n\}$ را به عنوان مجموعه‌ای از اندیس‌های تمامی بیت‌کانال‌ها در نظر می‌گیریم. همچنین $\mathcal{A}$ را به عنوان یک زیرمجموعه k عضوی از $\mathcal{I}_n$ در نظر می‌گیریم که اصطلاحاً مجموعه اطلاعات نامیده می‌شود. به علاوه $\mathcal{A}^c$ یک زیرمجموعه $(n-k)$ بیتی از $\mathcal{I}_n$ و مکمل زیرمجموعه $\mathcal{A}$ است که اصطلاحاً مجموعه بدون تغییر (ثابت) نامیده می‌شود. بردار ورودی $u = (u_1, u_2, \ldots, u_n) = (u_{\mathcal{A}}, u_{\mathcal{A}^c})$ شامل دو زیربردار است: (الف) بردار اطلاعات $u_{\mathcal{A}} = (u_i, i \in \mathcal{A})$ که یک زیربردار k بیتی است؛ (ب) بردار ثابت $u_{\mathcal{A}^c} = (u_i, i \in \mathcal{A}^c)$ 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 حالی که بردار ثابت متشکل از مقادیر ثابتی است که برای کدبردار حذف متوالی (زیربخش ۱.۳) از قبل شناخته شده هستند. ماتریس مولد $G_{\mathcal{A}}$ را با انتخاب k سطر از ماتریس $G_n = \mathcal{F}^{\otimes m}$، که اندیس‌هایشان متناظر با اندیس‌های بیت‌کانال‌های مجموعه اطلاعات $\mathcal{A}$ (متناظر با کمترین مقدار ممکن پارامترهای باتاکاریا) هستند، می‌سازیم. در واقع به ازای بیت کانال $W_n^{(i)}$، iامین سطر 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 $\mathcal{A}^c$ (متناظر با بیشترین مقدار ممکن پارامترهای باتاکاریا) هستند را می‌سازیم. 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 $R = |u_{\mathcal{A}}|/|x| = |\mathcal{A}|/n$ تعریف می‌شود که می‌توان آن را با انتخاب اندازه مجموعه اطلاعات $\mathcal{A}$ تنظیم کرد. بردار x را به عنوان کلمه‌کد n بیتی که از n بیت‌کانال با اندیس‌های $\mathcal{I}_n = \{1, 2, \ldots, n\}$ عبور می‌کند را در نظر می‌گیریم. همچنین بردار y را به عنوان بردار خروجی متناظر که به وسیله الگوریتم حذف متوالی (SC [7]) کدبرداری می‌شود را در نظر می‌گیریم. هدف اصلی کدبردار SC، تخمین بردار ورودی با آگاهی از مجموعه اطلاعات $\mathcal{A}$، بردار ثابت $u_{\mathcal{A}^c}$ و بردار خروجی کانال y است. در واقع، بیت‌های بردار ورودی در کدبردار SC به صورت رابطه (۱) تخمین زده می‌شوند.

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

[1] Binary input Discrete Memoryless Channel [2] Binary Erasure Channel [3] Binary Symmetric Channel [4] channel polarization [5] bit-channel [6] Bhattacharyya parameter

[7] Successive Cancelation

که در آن توابع تصمیم‌گیری $h_i : \mathcal{Y}^n \times X^{i-1} \to X,\ i \in \mathcal{A}$ برای همه $y_1^n \in \mathcal{Y}^n,\ \hat{u}_1^{i-1} \in \mathcal{X}^{i-1}$ به صورت رابطه (۲) محاسبه می‌شوند،

$$h_i(y_1^n, \hat{u}_1^{i-1}) \triangleq \begin{cases} 0, & \text{اگر } \frac{W_n^{(i)}(y_1^n, \hat{u}_1^{i-1}|0)}{W_n^{(i)}(y_1^n, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{در غیر این صورت} \end{cases} \quad (۲)$$

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

۱.۳ ساخت ماتریس توازن‌آزما کدهای قطبی

پارامترهای $n = 2^m$، $m \geq 1$ و $F = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ را در نظر می‌گیریم. با فرض $R < I(W)$ و $k = nR$، ماتریس توازن‌آزما H برای کد قطبی (n, k) از طریق مراحل زیر ساخته می‌شود [۱۰]:

- ابتدا حاصلضرب کرونکر mام ماتریس F که ماتریس $G_n = F^{\otimes m}$ به ابعاد $n \times n$ نتیجه می‌دهد را در نظر می‌گیریم.
- سپس ستون‌های ماتریس G_n را از چپ به راست به صورت $i = 1, 2, \ldots, n$ شماره‌گذاری می‌کنیم.
- پارامترهای باتاکاریا تمامی n بیت‌کانال‌ها

$$\{W_n^{(i)}, 1 \leq i \leq n\}$$

از طریق فرمول بازگشتی (۳) با مقدار اولیه $Z_{1,1}$ به دست می‌آوریم و به صورت $Z_n = (Z(n, i), 1 \leq i \leq n)$ مرتب می‌کنیم. نکته اینکه در رابطه بازگشتی (۳) چنانچه کانال W، $\mathrm{BEC}(\epsilon)$ باشد، مقدار اولیه $Z_{1,1}$ برابر با ϵ است.

$$Z_{2k,i} = \begin{cases} 2Z_{k,i} - Z_{k,i}^2 & 1 \leq i \leq k \\ Z_{k,i-k}^2 & k+1 \leq i \leq 2k \end{cases} \quad (۳)$$
$$k = 1, 2, 2^2, \ldots, 2^{m-1}$$

- جایگشت $\pi_n = (i_1, i_2, ..., i_n)$ برای مجموعه اندیس‌های n بیت کانال، یعنی $\mathcal{I}_n = \{1, 2, \ldots, n\}$، را به دست می‌آوریم به نحوی که نامساوی $Z_{n,i_j} \leq Z_{n,i_k}, 1 \leq j < k \leq n$ 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.
- حال ترانهاده ماتریس توازن‌آزما یا همان H^T را با انتخاب $n-k$ ستون از ماتریس $G_n = F^{\otimes m}$، که اندیس‌هایشان متناظر با اندیس‌های بیت‌کانال‌های مجموعه ثابت $\mathcal{A}^c$ (متناظر با بیشترین مقدار ممکن پارامترهای باتاکاریا) هستند، می‌سازیم. در واقع به ازای بیت‌کانال $W_n^{(i)}$، iامین ستون ماتریس G_n را انتخاب می‌کنیم.

تاکنون از کدهای قطبی در چندین طرح رمزنگاری کلید عمومی و کلید خصوصی استفاده شده است و قابلیت‌های مناسبی در این حوزه چه از نظر سطح امنیت و چه از نظر سطح کارآیی از خود نشان داده‌اند [۱۱-۱۳]. در ادامه این مقاله به ارائه یک طرح شناسایی مبتنی بر کدهای قطبی می‌پردازیم و سطح امنیت و کارآیی آنرا نیز بررسی می‌کنیم.

۴ طرح شناسایی پیشنهادی مبتنی بر کدهای قطبی (Id-PC)

در این بخش طرح شناسایی پیشنهادی مبتنی بر کدهای قطبی (Id-PC[1]) را توضیح می‌دهیم. طرح شناسایی Id-PC بر مبنای طرح شناسایی Stern می‌باشد که در آن به منظور کاهش حجم داده‌های عمومی از کدهای قطبی به جای کدهای تصادفی استفاده کرده‌ایم. بر اساس نکات مطرح شده در بخش ۳، در کدهای قطبی می‌توان ماتریس توازن‌آزمای کد را از ماتریس آشکار G_n بدست آورد. در این طرح به منظور ساخت ماتریس توازن‌آزما، می‌توان $n-k$ ستون از G_n را به صورت تصادفی انتخاب کرد. بنابراین با داشتن یک رشته n بیتی به وزن $n-k$ می‌توان ماتریس توازن‌آزمای کد قطبی را بدست آورد. همچنین در طرح شناسایی Id-PC به منظور کاهش هزینه ارتباطات بجای ارسال سه چکیده در هر دور، یک چکیده ارسال می‌شود. در مقاله [۵] نویسندگان روشی را برای کاهش تعداد چکیده ارسالی پیشنهاد می‌دهند که بر اساس این واقعیت است که، وارسی‌کننده می‌تواند دو تا از سه چکیده را در هر دور بازیابی کند. اثبات‌کننده طبق معمول سه چکیده c_1 و c_2 و c_3 را محاسبه می‌کند. اما پس از آن به جای ارسال هر سه این مقادیر، $h(c_1, c_2, c_3)$ را ارسال می‌کند. پس از دریافت چالش از وارسی‌کننده، اثبات‌کننده می‌داند که وارسی‌کننده دو تا از چکیده‌ها را می‌تواند محاسبه نماید. لذا اثبات‌کننده به طور معمول پاسخ می‌دهد، اما به پاسخ خود چکیده‌ای که وارسی‌کننده نمی‌تواند محاسبه نماید را نیز اضافه می‌کند. در مرحله تایید، وارسی‌کننده می‌تواند دو تا از چکیده‌ها را بازیابی نماید و چکیده سوم را نیز از اثبات‌کننده دریافت کرده است. حال می‌تواند اولین چکیده یعنی $h(c_1, c_2, c_3)$ را محاسبه نماید. به طور کلی در این روش بجای اینکه سه چکیده در هر دور ارسال شود، دو چکیده ارسال می‌شود. این ایده می‌تواند در مورد دورهای متوالی تعمیم یابد، در این حالت برای هر دور، اثبات‌کننده فقط مقدار چکیده‌ای که توسط وارسی‌کننده قابل بازیابی نیست را ارسال می‌کند و دو چکیده دیگر توسط وارسی‌کننده بازیابی می‌شود. در واقع، فقط یک چکیده کلی برای همه دورها ارسال می‌شود که چکیده‌ای از دنباله همه مقادیر چکیده در دورهای مختلف می‌باشد. این نقطه نظر بسیار کارآمد است، زیرا میانگین

[1] Identification Scheme based on Polar Codes (Id-PC)

تعداد چکیده ارسالی در هر دور از سه به یک کاهش می‌یابد، یعنی در هر دور بجای سه چکیده، یک چکیده ارسال می‌شود. این بدان معناست که در حالت چکیده ۱۶۰ بیتی، ۳۲۰ بیت در هر دور صرفه جویی می‌شود.

۱.۴ ساختار طرح شناسایی Id-PC

در طرح شناسایی Id-PC، اثبات‌کننده تمهیدات مربوط به تمام r دور را یکجا ایجاد می‌کند و سپس چکیده کلی تمام این مقادیر را محاسبه و ارسال می‌کند. وارسی‌کننده چالش‌های مربوط به r دور را برای اثبات‌کننده می‌فرستد و اثبات‌کننده نیز با توجه به چالش‌ها پاسخ‌های مناسب می‌دهد. وارسی‌کننده می‌تواند دو تا از چکیده‌ها را از پاسخ‌ها بازیابی نماید. آخرین چکیده توسط اثبات‌کننده به همراه پاسخ ارسال می‌شود. حال وارسی‌کننده قادر به محاسبه چکیده کلی است، که در اولین مرحله توسط اثبات‌کننده ارسال شده بود. با بررسی صحت آن پروتکل کامل می‌شود و احراز هویت تایید می‌شود. در نتیجه، در طرح پیشنهادی برای کاهش هزینه ارتباطات به جای ارسال سه چکیده در هر دور، فقط یک چکیده ارسال می‌شود. ساختار طرح شناسایی Id-PC به شرح ذیل می‌باشد:

داده های عمومی: پارامترهای $n, k, w \in \mathbb{N}$ و تابع چکیده ساز h و j یک بردار n بیتی به وزن $n-k$ که به صورت تصادفی انتخاب می‌شود و مشخص می‌کند که کدام یک از ستون‌های ماتریس آشکار G_n در H^T وجود دارد. هر درایه از j که یک باشد ستون معادل آن در G_n، جزء H^T می‌باشد. بدین ترتیب ماتریس توازن‌آزما H مربوط به کد قطبی (n,k) تولید می‌شود. r تعداد دور اجرای هر مرحله از الگوریتم می‌باشد.

کلید خصوصی: $s \in W_{۲,n,w}$ کلید n بیتی با وزن همینگ w

کلید عمومی: $S = Hs^T$

تمهیدات: اثبات‌کننده برای $i = ۱, \ldots, r$ به صورت تصادفی $y_i \in \mathbb{F}_۲^n$ و جایگشت π_i از $\{۱, \ldots, n\}$ را انتخاب می‌کند. سپس

$$c_{۱,i} = h(\pi_i, Hy_i^T), c_{۲,i} = h(\pi_i(y_i)), c_{۳,i} = h(\pi_i(y_i+s))$$

و همچنین

$$C = h(c_{۱,۱}, \ldots, c_{۱,r}, c_{۲,۱}, \ldots, c_{۲,r}, c_{۳,۱}, \ldots, c_{۳,r})$$

را محاسبه و در نهایت C را ارسال می‌کند.

چالش: وارسی‌کننده برای $i = ۱$ تا r به صورت تصادفی $b_i \in \{۰, ۱, ۲\}$ را انتخاب و $b = b_۱, \ldots, b_r$ را ارسال می‌کند.

پاسخ: برای $i = ۱$ تا r اثبات‌کننده مقادیر z_i را محاسبه می‌کند. اگر $b_i = ۰$ بود آنگاه $z_i = (y_i, \pi_i)$. اگر $b_i = ۱$ بود آنگاه $z_i = (y_i + s, \pi_i)$. اگر $b_i = ۲$ بود آنگاه $z_i = (\pi_i(y_i), \pi_i(s))$. حال $(z_۱, c_{۳-b_۱,۱}, \ldots, z_r, c_{۳-b_r,r})$ را برای وارسی‌کننده ارسال می‌کند.

تایید: در نهایت وارسی‌کننده درستی C را بررسی می‌کند. در واقع برای $i = ۱$ تا r وارسی‌کننده مراحل زیر را انجام می‌دهد. اگر $b_i = ۰$ بود، $c_{۲,i}$ و $c_{۱,i}$ را محاسبه می‌کند و $c_{۳,i}$ نیز برای وارسی‌کننده ارسال شده است. اگر $b_i = ۱$ بود، $c_{۳,i}$ و $c_{۱,i}$ را محاسبه می‌کند و $c_{۲,i}$ نیز برای وارسی‌کننده ارسال شده است. اگر $b_i = ۲$ بود، $c_{۲,i}$ و $c_{۳,i}$ را محاسبه می‌کند و بررسی می‌کند که $wt(\pi(s)) = w$ باشد. همچنین $c_{۱,i}$ نیز برای وارسی‌کننده ارسال شده است. حال وارسی‌کننده C را محاسبه می‌کند و در صورت صحت، اثبات‌کننده تایید می‌شود.

جدول ۱. فاکتور کار بدست آوردن کلید خصوصی s طرح شناسایی Id-PC به ازای nهای مختلف

طول قالب	$n = ۵۱۲$	$n = ۱۰۲۴$	$n = ۲۰۴۸$
فاکتور کار	$۲^{۶۱}$	$۲^{۱۲۲}$	$۲^{۲۴۵}$

۲.۴ پارامترهای طرح شناسایی Id-PC

در مقاله طرح شناسایی Stern [۲] توصیه شده است که طول کد n دو برابر بعد کد k باشد و از آنجا که در کدهای قطبی $n = ۲^m$ می‌باشد، بنابراین اندازه‌های ممکن طول کد و بعد کد (n,k) در نظر گرفته شده در طرح شناسایی Id-PC عبارتند از $(۱۰۲۴, ۵۱۲)$، $(۵۱۲, ۲۵۶)$ و $(۲۰۴۸, ۱۰۲۴)$. همچنین بهتر است وقتی $n = ۲k$ باشد، w کمی پایین‌تر از $۰/۱۱n$ باشد [۲]. در نتیجه مقادیر مناسب وزن همینگ، $w = ۵۶$ در حالت $n = ۵۱۲$، $w = ۱۱۰$ در حالت $n = ۱۰۲۴$ و $w = ۲۲۰$ در حالت $n = ۲۰۴۸$ است.

۵ تحلیل امنیت طرح شناسایی Id-PC

امنیت طرح شناسایی Id-PC وابسته به سختی مسئله کدبرداری سیندروم است که ثابت شده است یک مسئله NP-complete است. اگر وزن همینگ w کلید خصوصی s به اندازه کافی کوچک باشد، پیدا کردن s معادل یافتن کلمه کدی با وزن کم که در معادله $S = Hs^T$ صدق کند است. با توجه به تخمین‌های گفته شده در مقاله [۱۴]، فاکتور کار الگوریتم‌های شناخته شده که می‌تواند کلید خصوصی s را بدست بیاورد، بالاتر از $\mathcal{O}(\Omega^k)$ می‌باشد و Ω برای مقادیر انتخاب شده حدود ۱/۱۸ می‌باشد. به ازای $k = ۲۵۶$ این مقدار برابر $۲^{۶۱}$ است و به ازای $k =$ ۵۱۲ و $k = ۱۰۲۴$ این مقدار به ترتیب برابر با $۲^{۱۲۲}$ و $۲^{۲۴۵}$ است. جدول ۱ مقایسه فاکتور کار بدست آوردن کلید خصوصی s به ازای طول‌های مختلف کد قطبی در طرح شناسایی پیشنهادی Id-PC را نشان می‌دهد. در ادامه این بخش امنیت طرح شناسایی Id-PC را از سه دیدگاه احتمال تقلب، اثبات هیچ‌آگاهی و حمله کدبرداری مجموعه اطلاعاتی (ISD) بررسی می‌کنیم.

۱.۵ احتمال تقلب

در طرح شناسایی Id-PC یک فرد متقلب در صورتی که به دو تا از چالش‌ها پاسخ درست دهد می‌تواند طرح مذکور را جعل نماید. حال بایستی بررسی کنیم که فرد متقلب به چند تکرار نیاز دارد تا بتواند طرح شناسایی Id-PC را جعل کند. در ادامه سه روش جعل مشخص شده است:

- شخص متقلب y و π و $x \in \mathbb{F}_2^n$ (لزوما نه با وزن کم) را به صورت تصادفی انتخاب می‌کند، به گونه‌ای که

$$Hx^T = Hy^T + S$$

باشد، سپس $c_1 = h(\pi, Hy^T)$ و $c_2 = h(\pi(y))$ را طبق معمول و $c_3 = h(\pi(x))$ محاسبه می‌کند و ارسال می‌کند. در پاسخ به چالش اگر $b = 0$ بود، زوج (y, π) و اگر $b = 1$ بود، زوج (x, π) را می‌فرستد. اگر $b = 2$ باشد روش جعل نخست با شکست مواجه می‌شود.

- شخص متقلب y و π و $x \in W_{2,n,w}$ را به صورت تصادفی انتخاب می‌نماید، سپس $c_1 = h(\pi, Hy^T)$ و $c_2 = h(\pi(y))$ را طبق معمول و $c_3 = h(\pi(y, x))$ را می‌سازد و ارسال می‌کند. در پاسخ به چالش اگر $b = 0$ بود (y, π) و اگر $b = 2$ بود $(\pi(y), \pi(x))$ را می‌فرستد. اگر $b = 1$ باشد، روش جعل دوم با شکست مواجه می‌شود.
- شخص متقلب y و π و $x \in W_{2,n,w}$ را به صورت تصادفی انتخاب می‌نماید، سپس $c_1 = h(\pi, H(y+x)^T + S)$ و $c_2 = h(\pi(y))$ و $c_3 = h(\pi(y+x))$ را می‌سازد و ارسال می‌کند. در پاسخ به چالش اگر $b = 1$ بود $(y+x, \pi)$ و اگر $b = 2$ بود $(\pi(y), \pi(x))$ را می‌فرستد. اگر $b = 0$ باشد، روش جعل سوم با شکست مواجه می‌شود.

در کل احتمال تقلب دقیقا برابر با $\frac{2}{3}$ است. این بدان معناست که یک اثبات‌کننده حقیقی برای پذیرش، باید بارها و بارها پروتکل را تکرار کند. بنابراین برای احراز هویت باید پروتکل چندین بار تکرار شود و اگر پروتکل را r بار تکرار کنیم، احتمال تقلب برابر $(\frac{2}{3})^r$ می‌شود. مثلا برای احتمال تقلب 2^{-16} باید پروتکل را ۲۸ بار تکرار کرد زیرا $(\frac{2}{3})^{28} = 2^{-16}$. حال با توجه به سطح امنیت مورد نیاز طرح شناسایی Id-PC، تعداد تکرار r مشخص می‌شود. برای مثال برای رسیدن به احتمال تقلب کمتر، یعنی 2^{-32}، در طرح شناسایی Id-PC نیاز به ۵۵ بار تکرار داریم.

۲.۵ اثبات هیچ‌آگاهی

یک طرح شناسایی هیچ‌آگاهی، پروتکلی است که اجازه می‌دهد اثبات‌کننده به وارسی‌کننده ثابت کند که او دارای اطلاعات مخفی است، بدون اینکه معلوم شود اطلاعات مخفی چیست. فرض کنید که اثبات‌کننده می‌خواهد دارا بودن اطلاعات مخفی s را به وارسی‌کننده ثابت کند. وارسی‌کننده به یک کلید عمومی pk و داده‌های عمومی D دسترسی دارد. برای شروع، اثبات‌کننده داده تصادفی y را انتخاب می‌کند و $Y = f(y)$ را برای وارسی‌کننده می‌فرستد. در نظر داشته باشید که f یک تابع یک طرفه یا یک تابع چکیده‌ساز است. سپس وارسی‌کننده یک چالش تصادفی c را انتخاب کرده و برای اثبات‌کننده می‌فرستد. اثبات‌کننده بعد از دریافت c، z را بر اساس s، c و y محاسبه و ارسال می‌کند. در نهایت وارسی‌کننده با کمک pk و D، صحت z را بررسی می‌کند. اثبات اینکه هیچ اطلاعات خصوصی رد و بدل نشده، در طرح شناسایی Id-PC به سادگی امکان‌پذیر است. تنها اطلاعاتی که در طی اجرای پروتکل آشکار می‌شود، دو متغیر تصادفی y و π و رشته‌های جایگشتی $\pi(y)$ و $\pi(s)$ و نهایتا $y + s$ است. همچنین واضح است که y و π و $\pi(y)$ هیچ اطلاعاتی از s نمی‌دهند و رشته جایگشتی $\pi(s)$ بجز وزن همینگ s چیز دیگری را مشخص نمی‌کند (که از قبل نیز مشخص بوده است). همچنین $y + s$ مانند یک مقدار یکبار مصرف عمل می‌کند. دلیلش اینست که y به صورت تصادفی انتخاب می‌شود، از این رو به طور متوسط $wt(y) = \frac{n}{2}$ خواهد بود و حاصل $y + s$ تصادفی می‌شود.

۳.۵ حمله کدبرداری مجموعه اطلاعات

الگوریتم کدبرداری مجموعه اطلاعات (ISD) یکی از قوی‌ترین الگوریتم‌هایی است که برای حل مسئله کدبرداری 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 در مقاله اولیه‌اش با الهام از مقاله [۱۵] مطرح شد. پس از او در کارهای متعددی بهبودهایی در این حمله داده شد [۱۶-۱۸]. یکی از قوی‌ترین انواع الگوریتم‌های ISD توسط Stern مطرح شد [۱۹]. پس از Stern نیز کارهای متعدد دیگری در این حوزه انجام شده است [۲۰-۲۲]، 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. Stern چینش خطا را به صورت دو قسمت pتایی با فرض l ستون تمام صفر در نظر گرفت و با استفاده از تناقض‌نمای روز تولد پیچیدگی را به دست آورد. احتمال موفقیت الگوریتم Stern در یافتن یک کلمه‌کد با وزن همینگ داده شده w و با شرایط ذکر شده، به صورت

$$P_S = \binom{k/2}{p}^2 \binom{n-k-l}{\omega-2p} / \binom{n}{w}$$

است. پیچیدگی هر بار اجرای الگوریتم به صورت

$$\mathcal{N}_{Cost} = \frac{1}{2}(n-k)^2(n+k) + 2pl\binom{k/2}{p} + \frac{2p(n-k)\binom{k/2}{p}^2}{2^l}$$

محاسبه می‌شود. در نهایت فاکتور کار حمله ISD برابر با

$$WF_{isd}(n, k, \omega) = \mathcal{N}_{Cost}/P_S$$

می‌شود [۱۶]. جدول ۲ سطح امنیت طرح شناسایی Id-PC در برابر حمله کدبرداری مجموعه اطلاعاتی به ازای طول‌های مختلف کد قطبی و پارامتر را با استفاده از الگوریتم Stern برای سه دسته پارامتر نشان می‌دهد.

جدول ۲. سطح امنیت طرح شناسایی Id-PC در برابر حمله کدبرداری مجموعه اطلاعاتی به ازای nهای مختلف

	$n = ۵۱۲$،	$n = ۱۰۲۴$،	$n = ۲۰۴۸$،
پارامتر	$k = ۲۵۶$،	$k = ۵۱۲$،	$k = ۱۰۲۴$،
	$w = ۵۶$	$w = ۱۱۰$	$w = ۲۰۶$
سطح امنیتی	$۲^{۶۹٫۶۷۰۴}$	$۲^{۱۲۶٫۸۰۸۱}$	$۲^{۲۲۷٫۸۷۳۰}$
(p, l)	(۳, ۲۲)	(۳, ۲۶)	(۵, ۴۴)

جدول ۳. مقایسه اندازه داده عمومی طرح‌های شناسایی Stern و Id-PC به ازای nهای مختلف بر حسب بیت

کد عمومی / طول اندازه داده	$n = ۵۱۲$	$n = ۱۰۲۴$	$n = ۲۰۴۸$
طرح شناسایی Stern	۱۳۱۰۷۲	۵۲۴۲۸۸	۲۰۹۷۱۵۲
طرح شناسایی Id-PC	۵۱۲	۱۰۲۴	۲۰۴۸

۶ تحلیل کارآیی طرح شناسایی Id-PC

در این بخش کارآیی طرح شناسایی Id-PC را از دو دیدگاه اندازه داده عمومی و هزینه ارتباطات بررسی می‌کنیم.

۱.۶ اندازه داده عمومی

استفاده از کد قطبی در ساختار طرح شناسایی Id-PC باعث کم شدن حجم داده‌های عمومی می‌شود. زیرا می‌توان بجای اینکه ماتریس توازن‌آزما $H_{(n-k)\times n}$ را ذخیره نماییم، یک بردار n بیتی را ذخیره نمود و با استفاده از روش ارائه شده در بخش ۳ ماتریس توازن‌آزما را از روی ماتریس G_n بدست آورد. در نتیجه حافظه مورد نیاز برای ذخیره آن مثلا از ۵۲۴۲۸۸ بیت به ۱۰۲۴ بیت کاهش یابد. جدول ۳ مقایسه اندازه داده عمومی (ماتریس H) در طرح‌های شناسایی Id-PC و Stern را بر حسب بیت نشان می‌دهد. همانطور که در این جدول مشاهده می‌شود، اندازه داده عمومی در طرح پیشنهادی به دلیل استفاده از خواص کدهای قطبی به طور قابل ملاحظه‌ای کاهش یافته است. نکته این که نسبت اندازه داده عمومی در طرح شناسایی Stern به اندازه داده عمومی در طرح پیشنهادی به مقدار $\frac{n}{۲}$ برابر است.

۲.۶ هزینه ارتباطات

هزینه ارتباطات برای طرح‌های شناسایی Stern و Id-PC به ترتیب از رابطه‌های (۴) بیان شده در [۷] و (۵) بدست می‌آید که در آن‌ها l_π طول جایگشت π و r تعداد دور اجرای پروتکل می‌باشد.

$$Comm_{\text{stern}} = r \times \left(۳ \cdot l_{hash} + \frac{۴ \cdot n + ۲ \cdot l_\pi}{۳} \right) \quad (۴)$$

$$Comm_{\text{Id-PC}} = r \times \left(l_{hash} + \frac{۴ \cdot n + ۲ \cdot l_\pi}{۳} \right) \quad (۵)$$

جدول ۴. طول جایگشت π (l_π) طرح شناسایی Id-PC

	$n = ۵۱۲$	$n = ۱۰۲۴$	$n = ۲۰۴۸$
طول جایگشت π (بیت)	۲۰۴	۳۲۰	۴۹۸

برای محاسبه هزینه ارتباطات در طرح شناسایی Id-PC فرض می‌کنیم که تابع چکیده‌ساز دارای خروجی ۱۶۰ بیتی باشد. همچنین فرض می‌کنیم که جایگشت π توسط روش گفته شده در [۸] تولید شود. طول جایگشت شبه تصادفی با استفاده از این روش برابر $۲ \times m \times ۲^m$ بیت می‌باشد که در آن $m = \frac{\lceil \log_۲ n \rceil}{۲}$ است، در نتیجه به ازای $n = ۱۰۲۴$ طول جایگشت برابر با ۳۲۰ بیت می‌شود. جدول ۴ طول جایگشت π در طرح شناسایی Id-PC را به ازای nهای مختلف را بر حسب بیت نشان می‌دهد.

در واقع، در طرح شناسایی Id-PC جهت کم کردن هزینه ارتباطات از دو روش به صورت زیر استفاده شده است:

تولید بذر: روش نخست، روش تولید بذر [۷، ۸] است که در آن جایگشت تصادفی π و بردار تصادفی y معمولا از بذرهای تصادفی تولید می‌شوند. اثبات‌کننده و وارسی‌کننده می‌توانند در یک الگوریتم مشترک برای تولید π و y از این بذرها به توافق برسند. معمولا فقط بذر بجای کل π یا کل y ارسال می‌شود. برای سطح امنیتی $۲^{۱۲۸}$ فقط ۱۲۸ بیت برای بذر جایگشت π و یا بذر بردار y ارسال می‌شود، بجای اینکه چند صد بیت فرستاده شود. رابطه (۶) هزینه ارتباطات طرح Id-PC را با در نظر گرفتن راهکار ارسال بذر بجای π و y نشان می‌دهد. در این رابطه l_s طول بذر می‌باشد که بسته به سطح امنیتی مورد نیاز مقدار آن متغیر است. برای مثال، برای سطح امنیتی $۲^{۱۲۸}$، $l_s = ۱۲۸$ و برای سطح امنیتی $۲^{۶۴}$، $l_s = ۶۴$ بیت می‌باشد.

$$Comm_{\text{Id-PC}(seed)} = r \times \left(l_{hash} + \frac{۲ \cdot n + ۴ \times l_s}{۳} \right) \quad (۶)$$

فشرده‌سازی: روش دوم کاهش هزینه ارتباطات، فشرده‌سازی کلمات کد با وزن کم است. از آنجا که s یک کلمه‌کد با وزن کم می‌باشد جایگشت $\pi(s)$ آن نیز دارای وزن کم است، لذا با استفاده از الگوریتم فشرده‌سازی هافمن می‌توان طول آن را به نصف کاهش داد. رابطه (۷) هزینه ارتباطات طرح شناسایی Id-PC را با در نظر گرفتن راهکار فشرده‌سازی نشان می‌دهد.

$$Comm_{\text{Id-PC}(compression)} = r \times \left(l_{hash} + \frac{n + n/۲ + ۴ \times l_s}{۳} \right) \quad (۷)$$

نتایج حاصل از روابط (۴) تا (۷) در جدول ۵ خلاصه شده‌اند. در این جدول هزینه ارتباطات (بر حسب بیت) در حالت $n = ۱۰۲۴$ برای ۵۵ دور و در حالت $n = ۵۱۲$ برای ۲۸ دور اجرای پروتکل محاسبه شده است.

جدول ۵. مقایسه هزینه ارتباطات طرح‌های Stern و Id-PC

پارامترها شناسایی / طرح		$n = ۵۱۲$، $r = ۲۸$، $l_s = ۶۴$	$n = ۱۰۲۴$، $r = ۵۵$، $l_s = ۱۲۸$
هزینه ارتباطات (بیت) در طرح شناسایی Stern		۳۶۳۶۳	۱۱۳۲۲۷
هزینه ارتباطات (بیت) در طرح شناسایی Id-PC	طرح اصلی	۲۷۴۰۳	۹۵۶۲۷
	با جایگزینی بذر	۱۶۴۲۷	۵۵۷۳۴
	با فشرده سازی	۱۴۰۳۸	۴۶۳۴۷

همانطور که در جدول ۵ مشاهده می‌شود، بهینه‌سازی‌های اعمال شده در طرح شناسایی Id-PC منجر به بیش از ٪۶۰ صرفه‌جویی در هزینه ارتباطات نسبت به طرح شناسایی Stern می‌شود. همچنین کمترین میزان هزینه ارتباطات در طرح شناسایی Id-PC مربوط به روش فشرده‌سازی می‌باشد.

۷ نتیجه‌گیری

در این مقاله یک طرح شناسایی مبتنی بر کدهای قطبی (Id-PC) ارائه شد. با بهره‌گیری از خواص کدهای قطبی، حجم داده‌های عمومی از ۵۲۴۲۸۸ بیت در طرح شناسایی Stern به ۱۰۲۴ بیت در طرح شناسایی Id-PC به ازای طول کد $n = ۱۰۲۴$، کاهش یافت. همچنین با استفاده از تکنیک‌های تولید بذر و فشرده‌سازی، هزینه ارتباطات از ۱۱۳۲۲۷ بیت در طرح شناسایی Stern به ۴۶۳۴۷ بیت در طرح شناسایی Id-PC کاهش یافت. جهت بررسی سطح امنیت طرح شناسایی Id-PC، اثبات هیچ‌آگاهی، احتمال تقلب و همچنین حمله ISD مرتبط با آن ارائه شد. فاکتور کار اعمال حمله کدبرداری مجموعه اطلاعات بر طرح پیشنهادی به ازای طول کد $n = ۱۰۲۴$، برابر با $۲^{۱۲۶۸۰۸۱}$ است که نمایانگر سطح امنیت مناسب طرح مذکور در برابر این حمله است. علاوه بر آن، نشان داده شد که برای دستیابی به احتمال تقلب $۲^{-۳۲}$، نیاز به ۵۵ بار تکرار پروتکل طرح شناسایی Id-PC داریم.

مراجع

[1] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[2] Jacques Stern. A new identification scheme based on syndrome decoding. In *Annual International Cryptology Conference*, pages 13–21. Springer, 1993.

[3] Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[4] Pascal Véron. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1997.

[5] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In *International Workshop on Selected Areas in Cryptography*, pages 171–186. Springer, 2010.

[6] Philippe Gaborit and Marc Girault. Lightweight code-based identification and signature. In *2007 IEEE International Symposium on Information Theory*, pages 191–195. IEEE, 2007.

[7] Carlos Aguilar, Philippe Gaborit, and Julien Schrek. A new zero-knowledge code based identification scheme with reduced communication. In *2011 IEEE Information Theory Workshop*, pages 648–652. IEEE, 2011.

[8] Arnaud Dambra, Philippe Gaborit, Mylene Roussellet, Julien Schrek, and Nicolas Tafforeau. Improved secure implementation of code-based signature schemes on embedded devices. *Cryptology ePrint Archive*, 2014.

[9] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory*, 55(7):3051–3073, 2009.

[10] Naveen Goela, Satish Babu Korada, and Michael Gastpar. On lp decoding of polar codes. In *2010 IEEE Information Theory Workshop*, pages 1–5. IEEE, 2010.

[11] Reza Hooshmand and Mohammad Reza Aref. Polar code-based secure channel coding scheme with small key size. *IET Communications*, 11(15):2357–2361, 2017.

[12] Reza Hooshmand, Masoumeh Koochak Shooshtari, and Mohammad Reza Aref. Pkc-pc: A variant of the mceliece public-key cryptosystem based on polar codes. *IET Communications*, 14(12):1883–1893, 2020.

[13] Reza Hooshmand, Mohammad Reza Aref, and Taraneh Eghlidos. Physical layer encryption scheme using finite-length polar codes. *Iet Communications*, 9(15):1857–1866, 2015.

[14] Florent Chabaud. Asymptotic analysis of probabilistic algorithms for finding short codewords. In *Eurocode'92*, pages 175–183. Springer, 1993.

[15] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[16] Jeffrey S Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.

[17] Pil Joong Lee and Ernest F Brickell. An observation on the security of mceliece's public-key cryptosystem. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 275–280. Springer, 1988.

[18] Carlisle M Adams and Henk Meijer. Security-related comments regarding mceliece's public-key cryptosystem. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 224–228. Springer, 1987.

[19] Jacques Stern. A method for finding codewords of small weight. In *International colloquium on coding theory and applications*, pages 106–113. Springer, 1988.

[20] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *International conference on the theory and application of cryptology and information security*, pages 88–105. Springer, 2009.

[21] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(20.054n)$. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 107–124. Springer, 2011.

[22] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1+ 1= 0 improves information set decoding. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 520–536. Springer, 2012.