هفتمین کنفرانس بین المللی انجمن رمز ایران
۲۴ و ۲۵ شهریورماه ۱۳۸۹، دانشگاه صنعتی خواجه نصیرالدین طوسی
دانشگاه صنعتی خواجه نصیرالدین طوسی
انجمن رمزایران

تحلیل رفتار تفاضلی ساختار غیرمتقارن تابع فیستل بهبود حملات اکتشافی حدس و تعیین به رمزهای دنباله‌ای

بهنام فهیم‌نیا[۱]، احسان کاظمی[۲]، ترانه اقلیدس[۳]

دانشگاه صنعتی شریف، دانشکده مهندسی برق[۱و۲]
{Fahimnia_behnam, Ehsankazemi }@ee.sharif.edu
دانشگاه صنعتی شریف، پژوهشکده الکترونیک[۳]
teghlidos@sharif.edu

چکیده

حمله‌ی حدس و تعیین یکی از حملات عام اعمال‌شده به رمزهای دنباله‌ای است. در راستای ساختار یافته کردن حمله‌ی حدس و تعیین به مولدهای دنباله تا کنون، الگوریتم‌های اکتشافی به نام‌های الگوریتم حدس و تعیین ساده(SGD)، الگوریتم حدس و تعیین پیشرفته(AGD) و الگوریتم حدس و تعیین اکتشافی (HGD) مطرح شده‌اند. در هر سه الگوریتم، مرحله‌ی حدس در قالب یافتن یک پایه‌ی حدس برای حالت داخلی یک مولد دنباله مطرح شده است. در این مقاله با ارائه‌ی یک رویکرد جدید در مرحله‌ی تعیین الگوریتم‌های SGD و AGD شاهد کاهش پیچیدگی در هر دو مرحله‌ی حدس و تعیین این الگوریتم‌ها خواهیم بود. از آن‌جا که هدف نهایی حمله‌ی حدس و تعیین و الگوریتم HGD به تبع آن، حرکت به سوی یافتن پایه‌ی کمینه بوده است، به منظور دستیابی به پایه‌ی کمینه، در این مقاله الگوریتم جدیدی با دو نگرش ارائه می‌شود. نتایج حاصل از اعمال این الگوریتم، که آن‌را الگوریتم حدس و تعیین اکتشافی مرکب(CHGD)نامیده‌ایم، حاکی از بهبود عملکرد آن در هر دو مرحله‌ی حدس و تعیین است.

واژه‌های کلیدی

رمز دنباله‌ای، حمله‌ی حدس و تعیین، الگوریتم‌های اکتشافی، الگوریتم حدس و تعیین ساده، الگوریتم حدس و تعیین پیشرفته.

۱- مقدمه

رمزهای دنباله‌ای دسته‌ای از رمزهای متقارن هستند که در آن‌ها به کمک یک کلید اصلی یک دنباله‌ی کلید اجرایی تولید می‌شود. کلید اصلی باید در اختیار هر دو طرف فرستنده و گیرنده قرار گیرد و با مکانیزمی مشابه و با استفاده از مولد دنباله‌ی شبه‌تصادفی، دنباله‌ی کلید اجرایی را تولید کند. رمزهای دنباله‌ای به دو نوع رمزهای دنباله‌ای مبتنی بر بیت و رمزهای دنباله‌ای مبتنی بر کلمه تقسیم می‌شوند. سرعت زیاد انجام محاسبات پردازنده‌های کنونی در میدان‌های بزرگتر، گرایش به رمزهای دنباله‌ای مبتنی بر کلمه را افزایش داده است. در این رمزها، که هم‌اکنون مبنای طراحی اکثر رمزهای دنباله‌ای هستند از کلمه‌های w بیتی (معمولاً w ۱۶ یا ۳۲ بیتی است.) برای حالت‌های داخلی مولد دنباله‌ی کلید اجرائی استفاده می‌شود.

در حمله‌ی حدس و تعیین[۳،۲،۱] با توجه به شناخت از ساختار مولد کلید اجرایی به دنبال یافتن حالت داخلی سیستم هستیم به گونه‌ای که با حدس تعدادی از مؤلفه‌های حالت داخلی، بقیه‌ی مؤلفه‌های داخلی سیستم بر مبنای آن به دست آید. در این صورت به دنباله‌ی کلید اجرایی در لحظات بعد دسترسی خواهیم داشت. اساس حمله‌ی حدس و تعیین، همان‌گونه که از نام آن برمی‌آید، بر پایه‌ی حدس تعدادی از مؤلفه‌های حالت داخلی سیستم و تعیین سایر حالت‌های داخلی از روی مؤلفه‌های حدس

زده‌شده است. یعنی پیچیدگی حمله به دو مسئله‌ی پیچیدگی حدس تعدادی از مؤلفه‌های حالت و تعیین سایر مؤلفه‌های حالت داخلی سیستم در یک لحظه‌ی خاص بر مبنای آن تقسیم می‌شود. به بیان دیگر مسئله‌ی حمله‌ی حدس و تعیین به مسئله‌ی یافتن یک مجموعه‌ی پایه به منزله‌ی قسمت حدس‌زده‌شده برای دستیابی به کل حالت داخلی سیستم بر مبنای فرضی که در مورد مسئله‌ی تعیین می‌کنیم تبدیل می‌شود. یعنی دو چالش **حدس** و **تعیین** پیش‌روی تحلیلگر قرار می‌گیرد. در مدل‌های ارائه شده برای این حمله تاکنون [۵،۴،۳،۲،۱] مسئله‌ی اصلی نگرش هم‌زمان به این دو مرحله و ارائه‌ی الگوریتمی برای یافتن یک مجموعه‌ی پایه از عناصر حدس تعدادی از مؤلفه‌های حالت که منجر به یافتن حالت اولیه‌ی داخلی مولد دنباله‌ی کلید اجرائی در یک لحظه‌ی خاص شود، بوده است. زیرا یافتن پایه‌ی کمینه برای یک چنین سیستمی در حالت کلی یک مسئله‌ی دشوار است. پس از یافتن یک زیرمجموعه از مؤلفه‌های حالتی که باید حدس زده شوند، سایر مؤلفه‌های حالت داخلی سیستم بر مبنای فرض فازِ تعیین به دست می‌آیند و با استفاده از این حالت داخلی دنباله‌ی کلید اجرایی تولید می‌شود و با دنباله‌ی کلید اجرائی به دست‌آمده توسط تحلیلگر مقایسه می‌شود و در صورت برابری این دو دنباله، حدس موردنظر به عنوان حدس درست انتخاب می‌شود. این فرایند تا یافتن حدس درست بر مبنای برابری دو دنباله‌ی مذبور ادامه خواهد یافت و پیچیدگی آن، با فرض چشم‌پوشی از پیچیدگی مرحله تعیین (که در اکثر این حملات [۳،۲،۱] فرض قابل‌قبولی است)، از مرتبه‌ی $O(2^{w\ n})$ است که در آن n، تعداد پایه‌های حدس‌زده‌شده برای حالت داخلی سیستم است. دلیل این امر جستجوی جامع فضای حالت‌های حدس‌زده‌شده است.

حملات حدس و تعیین، در اکثر موارد، دارای روندی ابتکاری هستند و این امر با توجه به نگرش هم‌زمان به هر دو مسئله‌ی حدس و تعیین و درجه‌ی آزادی زیاد این حمله توجیه‌پذیر است. به عنوان مثال، حمله‌ی حدس وتعیین ساده بر مبنای الگوریتم شطرنجی [۱]، حمله‌ی حدس و تعیین پیشرفته [۲] و حمله‌ی حدس و تعیین اکتشافی [۳] نمونه‌ای از کوشش‌هایی بوده‌اند که با تکیه بر روش‌های اکتشافی و ابتکاری به حل این مسئله مبادرت ورزیده‌اند. در این مقاله، پس از بیان اجمالی این حملات، با اعمال تغییری در الگوریتم حدس و تعیین ساده (SGD)و الگوریتم حدس و تعیین پیشرفته (AGD) بهبودی در عملکرد آن‌ها داده خواهد شد. سپس، حمله‌ی حدس و تعیین اکتشافی (HGD) به طور کامل مطالعه می‌شود و بهبودی در آن به منظور تکمیل و افزایش کارایی آن داده خواهد شد. از آن‌جایی که الگوریتم قبلی هیچ ضمانتی برای دستیابی به یک مجموعه‌ی پایه‌ی کمینه ارائه نمی‌دهد، تلاش در راستای یافتن تعداد پایه‌های کمتر و حرکت به سمت دستیابی به پایه‌ی کمینه که هدف تغییر اعمال‌شده در الگوریتم حدس و تعیین اکتشافی بوده است کاملاً ضروری به نظر می‌رسد. فرض اساسی برای مرحله تعیین حملات ذکرشده این است که یک مؤلفه‌ی حالت داخلی سیستم رمز تنها در صورتی به دست می‌آید که سایر مؤلفه‌های موجود در آن معادله‌ی خاص، مشخص باشند. این فرض به دلیل ساختار معادلات است که لزوماً خطی نیستند. البته همان‌طور که واضح است این فرض بدبینانه است و اگر معادله‌ی خطی‌ای یافت شود، پیچیدگی کلی الگوریتم کاهش خواهد یافت.

با اعمال تغییری که در الگوریتم‌های SGD و AGD انجام شد، پیچیدگی مرحله حدس به میزان $O(2^{5.76})$ کاهش یافت. در حالی که هم‌زمان به میزان ۱۰،۲ درصد کاهش در زمان اجرای مرحله تعیین حاصل شد که در کل یک بهبود بسیار خوب را نشان می‌دهد. به طور معمول عملکرد این دو قسمت در راستای عکس هم است و اعمال تغییر در مرحله تعیین می‌تواند موجب افزایش پیچیدگی مرحله حدس شود و بالعکس. ولی تغییر اعمال‌شده در این قسمت به نوعی است که با تعداد بررسی کم‌تر و البته به قدر کفایت موجب کاهش پیچیدگی هر دو مرحله می‌شود. الگوریتم جدیدی که بر مبنای الگوریتم HGD در این مقاله معرفی می‌شود، CHGD، به طور متوسط کاهشی از مرتبه‌ی $O(2^{18})$ در پیچیدگی مرحله حدس نسبت به الگوریتم HGD دارد که هزینه‌ی این بهبود، دو برابر شدن زمان اجرای مرحله تعیین است که در قبال کاهش ذکرشده در پیچیدگی مرحله حدس قابل چشم‌پوشی است. در پایان باید اشاره کرد که این نتایج از رمزهای تصادفی به دست آمده‌اند و یک معیار کلی برای نشان دادن روند صحیح فرضیات متعدد مطرح‌شده در مقاله هستند و حملات به سیستم‌های واقعی که می‌تواند موضوع کارهای آینده باشد، محدوده‌ی کاربردی و عملی این الگوریتم‌ها را تأیید خواهند کرد.

سایر بخش‌های مقاله به ترتیب زیر ارائه می‌شود. در بخش ۲، حملات حدس و تعیین ساده [۱]، حدس و تعیین پیشرفته [۲] و حدس و تعیین اکتشافی [۳] مرور خواهند شد. در بخش بعد با اعمال تغییراتی در الگوریتم‌های بخش ۲ بهبودی در آن‌ها حاصل خواهد شد. در بخش ۴ الگوریتم حدس و تعیین ابتکاری تکمیل می‌شود. سپس مقایسه‌ی جامعی بین الگوریتم حمله‌ی بهبودیافته و سایر حملات ذکرشده در بخش ۲ به عمل می‌آید.

۲-حمله‌های حدس و تعیین پیشین!

۲-۱ حمله‌ی حدس و تعیین ساده(SGD)

این حمله[۱] کوششی برای یافتن یک پایه‌ی فضای حالت به منظور ساختار یافته کردن حمله‌ی حدس و تعیین به رمزهای دنباله‌ای است. بدنه‌ی اصلی این حمله یک الگوریتم به نام الگوریتم شطرنجی است که از یک زیرالگوریتم به نام الگوریتم اصلی

استفاده می‌کند. پیش از اعمال الگوریتم، نیاز به استخراج یک دسته معادلات (خطی یا غیرخطی) از سیستم رمز داریم. این معادلات از ساختار داخلی رمز موردنظر (تابع پس‌خورد ثبات انتقال خطی و جزء غیرخطی مولد دنباله‌ی کلید) به دست می‌آیند و هر معادله در طول فرمان‌های ساعت متوالی تکرار می‌شود و معادلاتی به شکل زیر به دست می‌آیند [۳]،

$$\begin{aligned} &f_1(S_{1,1,t+\Delta}, S_{1,2,t+\Delta}, \ldots, S_{1,k_1,t+\Delta}) = 0 \\ &f_2(S_{2,1,t+\Delta}, S_{2,2,t+\Delta}, \ldots, S_{2,k_2,t+\Delta}) = 0 \qquad 0 \le \Delta < n \quad (1) \\ &\vdots \\ &f_3(S_{3,1,t+\Delta}, S_{3,2,t+\Delta}, \ldots, S_{3,k_3,t+\Delta}) = 0 \end{aligned}$$

در حالت کلی، n حداقل برابر با تعداد مؤلفه‌های حالت داخلی سیستم است تا این اطمینان را ایجاد کند که کل حالت داخلی سیستم می‌تواند بازیابی شود. [۳] تعداد کل این معادلات برابر با $r \times n$ تا خواهد بود. که در آن، r تعداد معادلات دستگاه بالاست. این دستگاه معادلات برای سیستم‌های مولد کلید دنباله‌ای با فرمان ساعت منظم برقرار است و این فرض اساسی، که پایه‌ی حملات حدس و تعیین مذکور است، در سیستم‌های موردحمله باید وجود داشته باشد. در مورد سیستم‌های مولد کلید با فرمان ساعت نامنظم، چون اطلاع دقیقی از پالس‌زنی سیستم نداریم، مدل ارائه‌شونده‌ی حمله احتمالی خواهد بود. یعنی هر معادله‌ی موجود بین مؤلفه‌های حالت داخلی سیستم با یک احتمالی برقرار خواهد بود که کمتر از یک است. بنابراین کارآیی حمله‌ی حدس و تعیین بر مبنای این مدل بسیار کاهش خواهد یافت ولی به هر حال، صرف‌نظر از مسئله‌ی پیچیدگی، با یک مدل احتمالی می‌توان چنین حملاتی را به آن‌ها اعمال کرد. فرض اساسی مرحله تعیین این است که مؤلفه‌ی حالت موردنظر تعیین می‌شود، اگر تمامی مؤلفه‌های دیگر حالت در آن معادله تعیین شده باشند.

هر دستگاه معادلات به صورت (۱) معادل یک دسته ماتریس r تایی (یک ماتریس برای هر معادله) می‌تواند باشد که هر درایه‌ی آن ماتریس، شماره‌ی اندیس‌های حالتی است که در هر معادله حضور دارند. یعنی در سطر j ام ماتریس i ($1 \le i \le r$)، اندیس‌هایی که در هر معادله حضور دارند با ۱ و بقیه‌ی درایه‌ها را با صفر پر می‌کنیم. این ماتریس‌ها را ماتریسِ اندیس متغیرهای حالت گویند. در این صورت فرض اساسی مرحله تعیین به این صورت خواهد بود که درایه‌ی k در سطر j ماتریس اندیس حذف می‌شود اگر تمامی متغیرهای (درایه‌های) دیگر در آن سطر مشخص (حذف) شده باشند. در آن صورت، این درایه از تمامی سطرهای ماتریس اندیس حذف خواهد شد.

اگر معادلات استخراجی از مولد دنباله‌ی کلید، فرض اساسی مرحله تعیین را برآورده نکنند، باید به دنبال فرض‌های ساده‌شونده‌ای در معادلات باشیم که آن‌ها را به صورت معادلات موردنظر درآورد. البته این فرض‌های اضافی معادل با یک هزینه‌ی محاسباتی خواهد بود که به طور کامل باید در نظر گرفته شود و در پیچیدگی کلی لحاظ شود.

۲-۱-۱ توصیف الگوریتم شطرنجی

در این الگوریتم، یک زیرالگوریتم به نام الگوریتم اصلی به طور متناوب در هر دور اجرا می‌شود تا بهترین مجموعه‌ی پایه برای فضای حدس به دست آید. پس از به دست آوردن تمامی ماتریس‌های اندیس، M_i ، $1 \le i \le r$، بهترین کاندیداها برای فضای حدس انتخاب می‌شوند. در [۱]، چهار معیار A، B، C و D برای انتخاب این نامزدها معرفی شده‌اند. در این قسمت، معیار D که مؤثرتر از سایر نواحی است معرفی می‌شود [۲] .

معیار D

در این معیار اولویت انتخاب یک درایه و به طور معادل مؤلفه‌ی متناظر با آن، به عنوان پایه به صورت زیر است [۱]،

۱) بتواند درایه‌های بیشتری از ماتریس‌های اندیس را حذف کند.

۲) بتواند سطرهای با درایه‌های حذف‌شده‌ی بیشتری را تولید کند.

دو معیار بالا به این ترتیب می‌توانند توجیه شوند که اولاً متغیری انتخاب شود که بیشترین حضور را در معادلات دارد (معیار ۱) و ثانیاً با حذف آن ماتریس‌های اندیس تنک‌تر شوند. (معیار ۲)

با این فرض‌ها الگوریتم اصلی به صورت زیر بیان می‌شود،

۱) نامزدها را بر مبنای معیار D انتخاب کن.

۲) درایه‌های انتخاب شده را از ماتریس‌های اندیس حذف کن.

۳) سطرهای با یک درایه‌ی نامعلوم را حذف کن.

۴) اگر کل درایه‌های ماتریس حذف نشده باشد، به مرحله‌ی ۱ بازگرد.

همان‌طور که ذکر شد، الگوریتم شطرنجی به طور متناوب در هر دور، از زیرالگوریتم اصلی به صورت زیر استفاده می‌کند تا نامزدهای نهایی را انتخاب کند،

۱) نامزدهایی را انتخاب کن که در صورت حذف آن‌ها از ماتریس‌های اندیس، خروجی الگوریتم اصلی به فضای حدس کوچکتری منجر شود.

۲) این نامزدها را از کل ماتریس‌های اندیس حذف کن.

۳) سطرهای با یک درایه‌ی نامعلوم را حذف کن.

۴) اگر هنوز درایه‌ای از هر یک از ماتریس‌ها حذف نشده باشد، به مرحله‌ی ۱ بازگرد.

۲-۲ حمله‌ی حدس و تعیین پیشرفته (AGD)

در این حمله، در فرض اساسی مرحله تعیین نسبت به الگوریتم SGD تغییر ایجاد می‌شود، به طوری که این مرحله بر مبنای حل دستگاه معادلات حل‌پذیر از مراتب بیشتر است[۲] . هدف از این

تغییر کاهش تعداد عناصر پایه‌ی حدس است. البته همان‌گونه که مشهود است هزینه‌ی این کار افزایش پیچیدگی در مرحله تعیین است و بنابراین با یک بده-بستان بین کاهش اندازه‌ی پایه‌های حدس و افزایش پیچیدگی مرحله تعیین مواجه هستیم[۲].

۲-۲-۱ الگوریتم پیدا کردن سیستم معادلات خطی

در این الگوریتم [۲]، تحلیلگر ابتدا به دنبال یافتن سطرهای با معادلات تک‌متغیره است، سپس به دنبال دومعادله و دو متغیر می‌گردد و تا جایی که در الگوریتم در نظر گرفته شده به دنبال دستگاه معادلات حل‌پذیر مراتب بیشتر می‌گردد. در پیاده‌سازی الگوریتم باید توجه داشت که اندازه‌ی بیشینه‌ی دستگاه معادلات بسته به کاربرد می‌تواند متفاوت باشد، چون انتخاب یک مقدار بزرگ برای این اندازه می‌تواند پیچیدگی الگوریتم را به اندازه‌ای افزایش دهد که حمله ناکارآمد شود. حال به توصیف الگوریتم می‌پردازیم[۲].

ورودی: ماتریس‌های اندیس، MAX (اندازه‌ی بیشینه‌ی دستگاه معادلات)

خروجی: پایه‌های حدس برای حمله‌ی حدس و تعیین

رویه:

۱. قرار بده $n=1$ (اندازه‌ی تعیین‌شده برای دستگاه معادلات).

۲. قرار بده $t=n$ (تعداد متغیرهای باقیمانده).

۳. قرار بده $flag=0$ (اگر $flag=1$ ، به این معنی است که یک دستگاه معادلات پیدا شده است).

۴. قرار بده $k=0$ (تعداد معادلات پذیرفته‌شده).

۵. برای هر سطر از ماتریس‌ها عملیات زیر را انجام بده:

۱،۵. قرار بده $w=$ تعداد متغیرهای تعیین‌نشده در سطر.

۲،۵. اگر $w\le t$ ، متغیرهای سطر را ذخیره کن و قرار بده $t=t-w$ و $k=k+1$.

۳،۵. اگر $k=n$، یک دستگاه از n معادله و n متغیر یافته‌ایم که اگر ماتریس رتبه‌تمام باشد، آن را حل کن، و قرار بده $flag=1$.

۶. اگر $flag=0$ ، قرار بده $n=n+1$.

۷. اگر $n<MAX$ برو به مرحله‌ی ۲.

همان‌گونه که دیده می‌شود با افزایش مرتبه‌ی معادلات هم‌زمان به دلیل افزایش ترکیب‌های مختلف معادلات از این مرتبه، پیچیدگی نیز افزایش می‌یابد در حالی که احتمال دست‌یابی به پایه‌ی با اندازه‌ی کوچکتر، بیشتر می‌شود. البته در [۲] برای سیستم رمز TIPSY نشان داده شده است که با افزایش اندازه‌ی MAX از یک میزان به بعد، بهبودی در اندازه‌ی پایه‌های به دست‌آمده حاصل نمی‌شود.

۲-۳ حمله‌ی حدس و تعیین اکتشافی

در این حمله، یک نگرش اکتشافی مبتنی بر یک روش پویای برنامه‌نویسی بیان شده است[۳]. در این حمله نیز مشابه با دو حمله‌ی پیشین، ابتدا باید معادلات خطی یا غیرخطی مولد دنباله موردِ حمله به دست آید. فرض اساسی مرحله‌ی تعیین این است که تمامی متغیرها باید هم‌اندازه باشند و یک متغیر در یک معادله تنها در صورتی تعیین می‌شود که سایر متغیرهای موجود در آن معادله تعیین شده و در دست باشند. در این روش تمامی متغیرهای موجود در معادلات به عنوان گره‌های دیاگرام ترلیس [۶] در نظر گرفته می‌شوند. سپس با استفاده از روشی مبتنی بر الگوریتم کدگشایی ویتربی [۷]، پایه‌ی موردنظر برای قسمت حدس‌زده‌شده به دست خواهد آمد. اساس کار به این صورت است که در هر مرحله، گره‌های متناظر با مسیرهای انتخاب‌شده بررسی می‌شوند که آیا تشکیل یک پایه برای حالت داخلی سیستم می‌دهد یا نه. الگوریتم زمانی پایان می‌یابد که گره‌های انتخاب‌شده تشکیل یک پایه برای حالت داخلی بدهند.

دستگاه معادلات (۱) را در نظر بگیرید که بر طبق آن ماتریس اندیس‌ها را می‌سازیم. به هر معادله می‌توان یک ماتریس اندیس متناظر کرد. هر اندیس متناظر با یک مؤلفه از حالت داخلی سیستم مولد دنباله است. هدف یافتن مجموعه‌ی اندیس‌هایی از این ماتریس‌هاست، به گونه‌ای که کل حالت داخلی سیستم در یک لحظه‌ی خاص به دست آید. مسئله‌ی پیدا کردن کوچکترین پایه در حالت کلی یک مسئله‌ی دشوار است و هر یک از روش‌های به کار رفته برای حمله‌ی حدس و تعیین دارای نکات ابتکاری برای حل این مسئله بوده‌اند. در حمله‌ی حدس و تعیین اکتشافی از یک الگوریتم شبه‌ویتربی برای یافتن کوچکترین پایه استفاده شده است. در بدنه‌ی این الگوریتم از معیارهای اولویت که معیارهای ابتکاری است و مبنای متریک‌گذاری در الگوریتم شبه‌ویتربی است، استفاده می‌شود.

معیارهای اولویت

زیرمجموعه‌ای دارای اولویت است که در صورت حدس زدنِ آن، تعداد درایه‌های (معادل اندیس‌های مؤلفه‌های حالت داخلی) بیشتری از ماتریس‌های اندیس حذف شود [۳]. در صورتی که چند کاندیدای مختلف موجود باشند، زیرمجموعه‌ای انتخاب می‌شود که تعداد ماتریس‌های با سطرهای دومتغیره، سه‌متغیره و بیشتر را بیشینه کند. یعنی ماتریس را تنک‌تر کند و در صورت برابری تمام این حالات، یک زیرمجموعه به تصادف انتخاب می‌شود.

۲-۳-۱ الگوریتم شبه‌ویتربی

در این حمله، بر مبنای معیارهای اولویت، با استفاده از الگوریتم شبه‌ویتربی که مبتنی بر الگوریتم ویتربی [۷] است، مبادرت به

یافتن پایه‌ی حالت داخلی می‌شود [۳]. در این الگوریتم اکتشافی، هر درایه‌ی موجود در ماتریس اندیس‌ها معادل با یک گره در دیاگرام ترلیس است. تعداد کل این گره‌ها را برابر q در نظر بگیرید. دیاگرام، حداکثر دارای l مرحله است. که در آن، l بیانگر اندازه‌ی حالت داخلی سیستم مولد کلید است. هر مرحله معادل با انتخاب یک پایه برای حالت داخلی سیستم است. از میان گره‌های مجاور به گره i- ام، گره‌ای انتخاب می‌شود که بر مبنای معیارهای اولویت دارای برتری باشد. در پایان هر مرحله، پایه بودن مجموعه‌های انتخابی بررسی می‌شوند. الگوریتم به شرح زیر است،

ورودی: دیاگرام ترلیس و ماتریس‌های اندیس، M_i، $1 \le i \le r$.

خروجی: پایه‌های حدس برای حمله‌ی حدس و تعیین رویه:

۱. قرار بده $i = 1$ (شماره‌ی مرحله) و $flag = 0$.

۲. برای هر اندیس (s)، $0 \le s \le q$، مراحل زیر را انجام بده:

۱،۲. در بین تمامی مسیرهای ورودی به گره s در مرحله-ی i، مسیر با بیش‌ترین اولویت را به عنوان مسیر بازمانده انتخاب کن و بقیه‌ی مسیرها را حذف کن.

۲،۲. بررسی کن آیا گره‌های موجود در مسیر بازمانده، تشکیل پایه‌ی حدس برای فضای داخلی سیستم را می‌دهند. اگر پاسخ مثبت است، قرار بده $flag = 1$ و شماره‌ی گره‌ها را ذخیره کن.

۳. اگر $flag = 0$ و $i \le l-1$، قرار بده $i = i+1$ و به مرحله‌ی ۲ برو.

۴. اگر $flag = 1$ ، بهترین مجموعه‌ی پایه از بین (حداکثر) q مسیر بازمانده به طول $i+1$ را بر مبنای حداقل پیچیدگی حمله‌ی حدس و تعیین انتخاب کن.

۵. اگر $flag = 0$ ، کل فضای داخلی را به عنوان پایه انتخاب کن.(جستجوی جامع فضای حالت داخلی)

کران بالای پیچیدگی این الگوریتم از مرتبه‌ی $O(q^2 l)$ است که پیاده‌سازی آن را موجه می‌کند.

۳- بهبود در الگوریتم‌های SGD و AGD

در این بخش می‌خواهیم با اعمال تغییری در بدنه‌ی الگوریتم‌های SGD و AGD ، پیچیدگی آن‌ها را کاهش دهیم. این تغییر به این صورت است که جستجوی یک مجموعه‌ی پایه تا آن‌جا ادامه می‌یابد که یک حالت داخلی مولد دنباله‌ی کلید در یک لحظه‌ی مشخص به دست آید. به بیان دیگر مؤلفه‌های با اندیس‌های متوالی که بتوانند حالت داخلی را بپوشانند توسط پایه‌ی انتخاب‌شده مشخص شوند. در حالی که در الگوریتم حدس و تعیین ساده و پیشرفته این جستجو تا حذف کلیه‌ی درایه‌های ماتریس‌های اندیس ادامه می‌یابد. این تغییر از دو جنبه قابل بررسی است،

۱- به دلیل کوچک‌تر شدن فضای حالت‌هایی که قرار است تعیین شوند، امکان دست‌یابی به پایه‌های با اندازه‌ی کوچکتر فراهم می‌شود.

۲- حتی در صورت برابری اندازه‌ی پایه‌های حدس با الگوریتم‌های SGD و AGD، پیچیدگی زمانی الگوریتم کاهش می‌یابد. دلیل این امر نیز کوچک‌تر شدن فضای حالت‌هایی است که باید تعیین شوند.

پس به این منظور شرطی را به هر دو الگوریتم اضافه می‌کنیم که پس از انتخاب مؤلفه‌های حالت به عنوان پایه و به‌دست‌آوردن سایر مؤلفه‌های همان حالت، به بررسی این نکته بپردازد که آیا مؤلفه‌های حدس‌زده‌شده و نیز تعیین‌شده کل حالت داخلی را پوشش می‌دهند یا نه.

جدول ۱ نتایج اعمال الگوریتم‌های بهبودیافته را به ۲۰۰ مولد دنباله با ماتریس‌های اندیسی که یک سطر آن به طور تصادفی انتخاب شده است و بقیه‌ی سطرها از انتقال متوالی آن سطر تولید شده‌اند، نشان می‌دهد،

جدول ۱- مقایسه‌ی الگوریتم‌های حدس و تعیین تغییریافته ساده (MSGD) و پیشرفته (MAGD) با الگوریتم‌های SGD و AGD

نام الگوریتم	متوسط پایه‌های حدس	درصد بهبود پیچیدگی زمانی
SGD	۵,۴۴۵	-
AGD	۴,۷۴۵	-
MSGD	۵,۲۳۵	۸,۳۶
MAGD	۴,۴۷۰	۱۲,۳۲

هر یک از ماتریس‌های اندیس انتخاب‌شده از ۲ یا ۳ معادله که در ۲۴ پالس ساعت اجرا شده‌اند، تشکیل شده‌اند. ضمناً، MAX الگوریتم AGD، برابر با ۵ در نظر گرفته شده است. اگر مقایسه‌ای بین الگوریتم‌های SGD و AGD و الگوریتم‌های بهبودیافته MSGD و MAGD انجام دهیم، شاهد بهبود در هر دو مورد ذکرشده در ابتدای بخش خواهیم بود. دلیل این امر، همان‌گونه که ذکر شد، کوچک‌تر شدن فضای حالت‌هایی است که باید تعیین شوند. این بهبود در مورد دوم به طرز واضح‌تری نمایان است. به این معنی که علاوه بر بهبود نسبی در تعداد عناصر پایه‌ی حدس، به دلیل انجام دادن جستجوهای کم‌تر، به زمان اجرای کم‌تری نیاز است. این بهبودِ زمانی در الگوریتم AGD محسوس‌تر است. در صورت اعمال تغییر در الگوریتم AGD، نیازی به جستجو برای معادلات از مرتبه‌ی بیشتر احساس نمی‌شود و با پذیرفتن این

واقعیت که پیچیدگی جستجوی معادلات با افزایش مرتبه‌ی معادلات افزایش می‌یابد.

۴- بهبود حمله‌ی حدس و تعیین اکتشافی

۴-۱ بیان ضعف در الگوریتم HGD

الگوریتم شبه‌ویتربی در راستای کاهش تعداد عناصر پایه‌ی مورد نیاز برای استخراج حالت داخلی مولد دنباله پیاده‌سازی می‌شود. ولی هیچ ضمانتی برای دستیابی به یک مجموعه‌ی پایه‌ی کمینه ارائه نمی‌دهد. به بیان دیگر، یک کران بالا برای تعداد عناصر پایه‌ی موردنیاز را به دست می‌دهد. که بر مبنای آن می‌توان ادعا کرد که تعداد عناصر پایه‌ی (اندازه‌ی پایه) لازم برای بازیابی حالت داخلی سیستم (اندازه‌ی پایه‌ی کمینه) از اندازه‌ی پایه‌ی یافت‌شده در این الگوریتم فراتر نمی‌رود. برای بهبود این نقیصه می‌توان شرط زیر را بررسی کرد،

۱. بررسی اینکه آیا به کمک مؤلفه‌های حدس‌زده‌شده، حالت داخلی سیستم پوشش داده می‌شود به جای جستجو برای یافتن پایه‌هایی که کل درایه‌های موجود در ماتریس‌های اندیس‌ها را حذف کند. الگوریتم HGD در صورتی یک مجموعه را به عنوان پایه اعلام می‌کند که کل درایه‌های موجود در ماتریس‌های اندیس حذف شده باشند یا به عبارت دیگر مشخص شده باشند. در حالی که اگر حالت داخلی سیستم در یک لحظه‌ی مشخص یافت شود، برای مسئله‌ی ما کفایت می‌کند و می‌توان مجموعه‌ی حدس‌زده‌شده را به عنوان پایه اعلام کرد. این تغییر در الگوریتم شبه‌ویتربی کارایی آن را از دو نقطه‌نظر بهبود می‌دهد،

- تعداد پایه‌های به دست‌آمده حداکثر به تعداد پایه‌هایی خواهد بود که در الگوریتم اصلی شبه‌ویتربی یافت می‌شود.
- حتی در صورت برابری تعداد پایه‌های یافت‌شده توسط هر دو الگوریتم، زمان اجرای الگوریتم را پایین می‌آورد.

۴-۲ یافتن پایه با نگرش حذفی

می‌توان با نگرشی مستقل و مشابه با الگوریتم اصلی شبه‌ویتربی، یافتن پایه‌ها را با ضریب اطمینان بیشتری انجام داد که در بخش بعد شرح می‌دهیم. در الگوریتم اصلی ما به دنبال حدس مؤلفه‌هایی از یک حالت داخلی سیستم هستیم که بتوان بر مبنای آن‌ها حالت داخلی سیستم را تعیین کرد. یعنی در این حالت، چالش اصلی ما انتخاب است. می‌توان این انتخاب را از حذف تعدادی از مؤلفه‌های حالت که به عنوان پایه انتخاب نمی‌شوند انجام داد. یعنی در نگرش جدید، چالش اصلی ما حذف خواهد بود؛ و معیارهای اولویت را باید بر مبنای آن تغییر داد.

معیارهای اولویت تغییر یافته

در این حالت، زیرمجموعه‌ای دارای اولویت است که در صورت حدس زدن آن‌ها، تعداد درایه‌های (معادل اندیس‌های مؤلفه‌های حالت داخلی) کم‌تری از ماتریس‌های اندیس حذف شود. در صورتی که چند نامزد مختلف موجود باشند، زیرمجموعه‌ای انتخاب می‌شود که تعداد ماتریس‌های با سطرهای دومتغیره، سه‌متغیره و بیشتر را کمینه کند. یعنی ماتریس را چگال‌تر کند و در صورت برابری تمام این حالات، یک زیرمجموعه به تصادف انتخاب می‌شود.

از آن‌جایی که الگوریتم شبه‌ویتربی هیچ تضمینی برای یافتن پایه‌های کمینه نمی‌دهد و توجه به این نکته که نگرش جدید به پایه‌ی حذف حالت‌های نشدنی، مستقل از نگرش الگوریتم اصلی است، احتمال یافتن پایه‌ی کمینه را افزایش می‌دهد. به این صورت که اگر اندازه‌ی پایه‌ی یافت‌شده در الگوریتم اصلی را با N_1 و اندازه‌ی پایه‌ی یافت‌شده با اعمال نگرش حذفی را با N_2 نشان دهیم، کران بالای اندازه‌ی پایه‌ی کمینه به صورت به دست می‌آید،

$$\text{اندازه‌ی پایه‌ی کمینه} \le \min\{N_1, N_2\} \qquad (۲)$$

البته می‌توان هر دو تغییر ذکرشده را با هم اعمال کرد. به بیان دقیق‌تر پس از تغییر در روند الگوریتم بر مبنای نگرش حذفی، شرط مذکور بررسی شود. و در این صورت کران بالای اندازه‌ی پایه‌ی کمینه به صورت کمینه‌ی اندازه‌ی پایه‌ی به دست‌آمده در الگوریتم اصلی و الگوریتم تغییریافته با اعمال شرط مذکور خواهد بود. الگوریتم جدید را الگوریتم حدس و تعیین اکتشافی مرکب(CHGD) می‌نامیم. دلیل این انتخاب استفاده از دو روند مستقل است که هر دو بر مبنای الگوریتم ویتربی کار می‌کنند.

ذکر این مطلب ضروری است که تمامی این تغییرها حرکتی به سوی دستیابی به پایه‌ی کمینه است و نه لزوماً ارائه‌ی یک الگوزیتم برای یافتن پایه‌ی کمینه. البته اعمال تمامی این تغییرات، بنا به دلائل ذکرشده، باعث می‌شود که کارآیی الگوریتم جدید نسبت به الگوریتم شبه‌ویتربی بهتر شود. و نتایج شبیه‌سازی که در زیربخش۴-۳ آورده خواهد شد مؤید این مطلب است.

۴-۳ نتایج و مقایسه

مشابه روندی که در [۳] آمده است، سیستم رمز SOSEMANUK [۸]، با استفاده از الگوریتم CHGD مورد حمله قرار گرفت که پیچیدگی این حمله از مرتبه‌ی $O(2^{224})$ است. روند حمله کاملاً منطبق بر [۳] است و تنها تفاوت آن در اعمال الگوریتم CHGD به جای الگوریتمHGD روی ماتریس‌های اندیس سیستم رمز SOSEMANUK است.

جدول ۲ مقایسه‌ی الگوریتم CHGD را با سه الگوریتم SGD، AGD و HGD نشان می‌دهد. این الگوریتم‌ها روی ۵۰۰ مولد دنباله با ماتریس‌های اندیسی که یک سطر آن به طور تصادفی انتخاب شده است و بقیه‌ی سطرها از انتقال متوالی آن سطر تولید شده‌اند، اعمال شده است. این رمزهای تصادفی دارای ۲ و یا ۳ معادله‌ی مشخصه هستند که هر معادله در ۳۲ پالس ساعت اجرا شده است و ماتریس اندیس متناظر به دست آمده است. لازم به ذکر است که مقدار MAX در الگوریتم AGD، برابر با ۳ انتخاب شده است.

جدول ۲- مقایسه‌ی عملکرد الگوریتم CHGD با الگوریتم‌های SGD ، AGD و HGD در دو مرحله‌ی حدس و تعیین و معیار مرکب

الگوریتم	متوسط پایه‌ی یافت‌شده	زمان اجرا (نرمالیزه)	معیار ترکیبی(نرمالیزه)
SGD	۶,۴۵۲	۱	۴,۴۵۲
AGD	۵,۷۲۰	۱۷۸,۶۶	۰,۲۳۸
HGD	۵,۹۸۶	۳۹,۴۴	۱
CHGD	۵,۸۵۰	۷۴,۴۲	۰,۴۱۸

در حملات حدس و تعیین، همانگونه که قبلاً ذکر شد، پیچیدگیِ الگوریتم ناشی از هر دو مرحله‌ی حدس و تعیین است. از آن‌جایی که مبنای طراحی سیستم‌های رمز موجود طراحی مبتنی بر کلمه است، هر مؤلفه‌ی داخلی سیستم یک کلمه‌ی ۱۶ یا ۳۲ بیتی است. پس هر ۱ واحد کاهش در تعداد پایه‌های حدس زده شده، معادل کاهشی 2^{16} یا 2^{32} برابر در پیچیدگی است. در حالی که پیچیدگی زمانی مرحله‌ی تعیین به طور مستقیم در پیچیدگی کل تاثیرگذار است. بنابراین در جدول ۲، حاصل‌ضرب عدد مربوط به ستون سوم در 2^{16} به توان تعداد متوسط عناصر پایه که روی الگوریتم HGD نرمالیزه شده است به عنوان یک معیار مرکب در مورد موفقیت این الگوریتم به کار می‌رود. به عنوان مثال عدد مربوط به الگوریتم SGD در ستون چهارم به شکل زیر به دست می‌آید:

$$\frac{2^{16(6.452)}}{2^{16(5.986)}}\times\frac{1}{39.44}=4.45187\approx 4.452$$

همان‌گونه که مشاهده می‌شود الگوریتم مرکب عمل‌کرد بهتری نسبت به الگوریتم‌های ساده و اکتشافی، که آن‌ها نیز از معادلات مرتبه یک به عنوان معیار اولویت استفاده می‌کنند، دارد. ولی الگوریتم پیشرفته که به جستجوی معادلات از مراتب بیشتر می‌پردازد، با توجه به معیار ترکیبی عمل‌کرد بهتری از خود نشان می‌دهد. البته اگر مبنای ما پیچیدگی زمانی اجرای الگوریتم، مستقل از مرحله حدس باشد الگوریتم CHGD در قیاس با سایر الگوریتم‌ها عمل‌کرد بهتری دارد. این مبنای تصمیم‌گیری در مواردی که قدرت پردازش ما تحت تأثیر سیستم موردحمله قرار می‌گیرد، شایان توجه است.

۵- نتیجه‌گیری

حمله‌ی حدس و تعیین یکی از حملات عام به رمزهای دنباله‌ای است که عمدتاً به شکل ابتکاری و غیر ساختاریافته مدل‌سازی می‌شود. در این مقاله، دو حمله‌ی حدس و تعیین موجود یعنی، حمله‌ی حدس و تعیین ساده (SGD) و حمله‌ی حدس و تعیین پیشرفته (AGD) بررسی شدند و با اعمال تغییری در آن‌ها میزان موفقیتشان را افزایش داده‌ایم.

این بهبود از دو جنبه‌ی کاهش در تعداد متوسط عناصر پایه‌ی حدس به میزان ۴,۴ درصد، که معادل کاهش در پیچیدگی به میزان $O(2^{5.76})$ است، و کاهش در زمان اجرای الگوریتم به میزان ۱۰,۰۲ درصد، قابل توجه است.

در ادامه پس از مروری بر حمله‌ی HGD، که یک روش اکتشافی بر مبنای روش کدگشایی ویتربی است، آن‌را با ابداع یک روش مشابه و مستقل و ترکیب آن با الگوریتم HGD، به نام الگوریتم حدس و تعیین اکتشافی مرکب (CHGD)، به یک ابزار جدید و کامل‌تر برای حمله‌ی حدس و تعیین به رمزهای دنباله‌ای دست یافته‌ایم. این الگوریتم قادر به کاهش تعداد عناصر پایه‌ی حدس به میزان ۸,۹ درصد نسبت به الگوریتم HGD است. در حالی که هزینه‌ی اجرای آن تقریباً دو برابر این هزینه برای الگوریتم HGD است. نتایج اعمال این حمله به سیستم رمز دنباله‌ای SOSEMANUK دارای پیچیدگی از مرتبه‌ی $O(2^{224})$ است که با پیچیدگی بهترین حمله‌ی موجود به آن برابری می‌کند.

در پایان باید اشاره کرد که یافتن پایه‌های کمینه برای یک چنین دستگاه معادلات به‌دست‌آمده از سیستم‌های مولد دنباله(مورد استفاده در رمزهای دنباله‌ای)، که خاصیت جمع‌شوندگی(ترکیب دو معادله، لزوماً به یک معادله‌ی معتبر منجر نمی‌شود.) ندارد، یک مسأله‌ی باز است. ضمناً تغییراتی که در معیارهای اولویت و الگوریتم‌های حمله‌ی متنوع ارائه‌شده در این مقاله می‌تواند داده شود و هم‌چنین ترکیب آن‌ها، می‌تواند پایه‌گذار حملات حدس و تعیین جدید و مؤثرتری باشد که پیچیدگی حملات به رمزهای دنباله‌ای را کاهش دهد.

۶- سپاس‌گذاری

این مقاله توسط بنیاد ملی علمی ایران(INSF) و مرکز مخابرات ایران(ITRC)، به شماره قرارداد ۱۹۱۶۷/۵۰۰، حمایت شده است.

مراجع

[۱] محمدی چم‌بلبـل، آزاد، **"تحلیل رمزهای دنباله‌ای مبتنـی بـر کلمه"**، پایان‌نامه‌ی کارشناسی ارشد، دانشگاه صنعتی شریف، ایران، ۱۳۸۳.

[۲] احمدی، هادی، **"بررسی حملات حدس و تعیین به سیستم‌هـای رمـز دنبالـه‌ای اسـتاندارد NESSIE و ارائـه‌ی یـک طـرح بهبودیافته برای رمزهای دنباله‌ای با انتقال کلمه به کلمـه"**. پایان‌نامه‌ی کارشناسی ارشد، دانشگاه صنعتی شریف، ایران، ۱۳۸۴.

[3] Ahmadi H., Eghlidos T., "Heuristic guess-and-determine attacks on stream ciphers ", IET Journal in Information Security, Vol. 3, Issue 2, pp. 66-73, June 2009.

[4] Hawkes P., Rose G., "Guess and determine attack on SNOW", SAC 02, LNCS, 2595, pp. 37-46, 2002.

[5] Mattsson J, "A guess and determine attack on the stream cipher polar bear", SAC 06, Vol.2002/017, pp.149-153, 2006.

[6] Lin S., Costello D.J, "Error control coding: fundamentals and applications", 2nd Edition, NJ, Upper Saddle River, Pearson Prentice-Hall.

[7] Viterbi A.J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Trens. Inf. Theory, IT-13, pp. 260-269, 1976.

[8] Berbian C., Billet O., Canteaut A., "SOSEMANUK, a fast software-oriented stream cipher". eSTREAM, ECRYPT Stream Cipher Project Report 2005/027, 2005.