مجله علمی - پژوهشی "پدافند الکترونیکی و سایبری"
سال سوم، شماره ۳، پاییز ۱۳۹۴، ص ۵۷- ۴۹

انتخاب بهینه پارامترهای حمله Rainbow TMTO با در نظر گرفتن زمان شکست در دو حالت جستجوی ترتیبی و جستجوی نشانه‌گذاری‌شده

محمد هادی[۱*]، محمد معینی جهرمی[۲]
۱- دانشجوی دکتری مخابرات، دانشگاه صنعتی شریف
۲- مربی و عضو هیئت علمی دانشگاه پیام نور
(دریافت: ۹۳/۱۱/۱۳؛ پذیرش: ۹۴/۱۰/۲۲)

چکیده

میزان حافظه، زمان شکست و احتمال موفقیت، مهم‌ترین معیارهای دخیل در عملکرد حمله TMTO هستند. انتخاب مناسب پارامترهای حمله همانند تعداد و طول زنجیرها به‌گونه‌ای که مقادیر مناسب برای معیارهای مذکور به‌دست آید، از اساسی‌ترین چالش‌ها در حمله TMTO به شمار می‌رود. مقالات زیادی جهت انتخاب مناسب و بهینه این پارامترها ارائه‌شده است اما در غالب آن‌ها جهت ساده‌سازی مسئله از زمان شکست صرف‌نظر شده و پارامترهای حمله بدون در نظر گرفتن زمان شکست محاسبه و یا بهینه شده‌اند. در این مقاله، رویکردی تازه جهت انتخاب بهینه پارامترهای حمله TMTO از نوع Rainbow، با در نظر گرفتن زمان شکست ارائه شده است. برای این منظور مشخصات اصلی حمله TMTO در یک مسئله بهینه‌سازی تلفیق‌شده است. هدف این مسئله کمینه کردن میزان حافظه مورد نیاز مشروط بر احتمال موفقیت معین و حداکثر زمان شکست مشخص می‌باشد. از آنجا که نحوه جستجو هنگام اجرای حمله نیز در نحوه عملکرد حمله TMTO موثر است، عملیات بهینه‌سازی برای دو شیوه جستجوی متداول ترتیبی و نشانه‌گذاری‌شده، انجام می‌گردد. نتیجه نهایی مقاله، دو رابطه ریاضی برای انتخاب پارامترهای حمله Rainbow TMTO یعنی تعداد و طول زنجیرها به ازای هریک از شیوه‌های جستجوی ترتیبی و نشانه‌گذاری‌شده می‌باشد. در انتها، کاربرد رویه انتخاب بهینه پارامترهای حمله، با مثال نشان داده می‌شود.

واژه‌های کلیدی: حمله **TMTO**، زمان شکست، جستجوی ترتیبی، جستجوی نشانه‌گذاری‌شده، بهینه‌سازی مقید

۱- مقدمه

حمله TMTO به‌عنوان یک روش حمله میانه، مصالحه‌ای میان دو پارامتر اصلی حمله یعنی حافظه و زمان شکست ایجاد می‌کند. برای این منظور، تمام فضای رمزنگاری به‌صورت مناسب در سطرهایی به نام زنجیر دسته‌بندی می‌شوند و فقط نقاط ابتدایی و انتهایی زنجیر در حافظه ذخیره می‌گردند [۱]. در زمان حمله برای یافتن کلید موردنظر، جدول تولیدی جستجو می‌شود. ایده اولیه حمله TMTO توسط Hellman ارائه شد و سپس با بهبود این ایده اولیه، گونه‌های پیشرفته‌تری از این حمله ارائه شد. روش نقاط متمایز و شیوه جداول Rainbow که به ترتیب توسط Rivest و Oechslin معرفی شدند، از مشهورترین گونه‌های بهبودیافته حمله TMTO هستند [۲-۱]. در شیوه نقاط متمایز، تولید زنجیر تا حصول یک الگوی ویژه در نقطه انتهایی زنجیر ادامه می‌یابد و به‌تبع آن طول زنجیرهای تولیدی در این روش متفاوت هستند. میزان دسترسی به حافظه در روش نقاط متمایز نسبت به روش ساده Hellman کمتر بوده و به همین سبب زمان شکست نیز کاهش می‌یابد. در روش جداول Rainbow، با استفاده از توابع تصادفی ساز میزان وقوع عوامل نامطلوب در تولید زنجیرها کاهش‌یافته و به همین جهت احتمال موفقیت در این‌گونه از حمله TMTO نسبت به حمله TMTO ساده بیشتر است [۴-۳].

احتمال موفقیت در حمله TMTO تابعی از تعداد و طول زنجیرهای تولید شده است [۲-۱]. بنابراین، انتخاب مناسب این عوامل برای رسیدن به بازدهی موردنظر حیاتی است [۳]. بهینه‌سازی می‌تواند تنها ناظر به تشکیل جدول‌ها و بدون توجه به زمان شکست، انجام شده باشد. برای نمونه در مرجع [۴]، یک شیوه مناسب برای افزایش احتمال موفقیت حمله ساده TMTO ارائه‌شده است. اگرچه این بهینه‌سازی روشی برای انتخاب مقادیر بهینه طول و تعداد زنجیرها با هدف افزایش احتمال موفقیت به دست می‌دهد اما نقص آن، عدم توجه به زمان شکست است. در مرجع [۵] نیز یک رابطه بهینه میان هزینه شکست و زمان، معرفی‌شده است. باید توجه داشت که کاهش هزینه ساخت ادوات

رایانامه نویسنده پاسخگو: mhadi@ee.sharif.edu

دیجیتال، از اهمیت بهینه‌سازی با دخالت هزینه کاسته است. همچنین، برخی نویسندگان چندین روش مرتب‌سازی و جستجوی جدید برای بهبود حمله TMTO ارائه کرده‌اند [۶]. این روش‌ها تنها به تغییر ساختار عملیات حمله توجه داشته و ناظر به انتخاب بهینه عوامل دخیل در حمله نیستند.

در این نوشتار، یک روش جهت انتخاب بهینه پارامترهای حمله TMTO از نوع Rainbow ارائه می‌گردد. دلیل انتخاب روش Rainbow این است که پیچیدگی زمان این روش نسبت به روش‌های دیگر TMTO به نسبت 2 کاهش می‌یابد [۲، ۸- ۷]. ما بر روی زمان شکست تمرکز می‌کنیم و می‌کوشیم به ازای احتمال موفقیت مشخص و حداکثر زمان شکست معین، حداقل میزان حافظه موردنیاز را به‌دست آوریم. در این راستا، دو معیار زمان شکست و حافظه که دو رکن اساسی حمله TMTO هستند، در قالب یک مسئله بهینه‌سازی تلفیق می‌شوند. نوآوری این رویکرد ازآن‌جهت است که برخلاف فعالیت‌های مشابه که از زمان شکست به دلیل ساده‌سازی تحلیل مسئله صرف‌نظر شده است [۹]، زمان شکست از ارکان اصلی بهینه‌سازی این نوشتار به شمار می‌رود. علاوه بر این، برای جستجوی جدول در مرحله شکست، دو شیوه جستجوی نشانه‌گذاری شده و جستجوی ترتیبی در نظر گرفته می‌شود. اگرچه حل مسئله بهینه‌سازی تعریف‌شده نیازمند تحلیل‌های نسبتا پیچیده است اما نتیجه نهایی دو رابطه فشرده ریاضی است که مقادیر بهینه پارامترهای جداول Rainbow، یعنی تعداد و طول زنجیرها را مشخص می‌کند. در پایان باید توجه داشت که بهینه‌سازی این نوشتار به‌گونه‌ای مزایای سایر بهینه‌سازی‌های پیشین را تلفیق و ازاین‌جهت بر آن‌ها جامعیت دارد. این مزایا شامل توجه به زمان شکست، عوامل دخیل در تشکیل جداول و شیوه‌های مختلف جستجو هستند.

۲- مقدمه‌ای بر حمله Rainbow TMTO

روش TMTO، روش میانه‌ای بین شیوه‌های حمله جستجوی کامل (Brute-Force) و لغت‌نامه (Dictionary) است. در حمله جستجوی کامل، با فرض در اختیار داشتن یک جفت کلمه ساده (Plain) و کلمه رمز (Cipher) نظیر، کل فضای کلید برای دست‌یابی به کلید رمز جستجو می‌شود. اگرچه احتمال موفقیت در این شیوه یک است اما زمان و قدرت پردازش حمله‌کننده ممکن است انجام این حمله را غیرممکن سازد. از طرف دیگر در حمله لغت‌نامه، با فرض در اختیار داشتن یک کلمه ساده معین، کلمه رمز به ازای تمامی حالات کلید محاسبه و ذخیره می‌شود. اکنون با در اختیار داشتن یک کلمه ساده و کلمه رمز نظیر، می‌توان کلید مربوطه را با جستجو در جدول یافت. احتمال موفقیت در این شیوه نیز یک بوده و شکست به‌صورت آنی انجام می‌شود اما این امر با صرف حافظه بسیار برای ذخیره‌سازی داده‌ها به‌دست می‌آید. روش حمله TMTO مصالحه‌ای میان زمان و حافظه ایجاد می‌کند به‌صورتی‌که حافظه موردنیاز آن به اندازه حمله لغت‌نامه و زمان موردنیاز آن به میزان حمله جستجوی کامل نباشد [۲، ۸- ۷]. از میان نسخه‌های مختلف حمله TMTO، روش Rainbow بیشتر موردتوجه قرار گرفته است زیرا در روش Rainbow، هدر رفتن حافظه و زمان ناشی از عوامل نامطلوب (مانند ادغام و برخورد زنجیرها)، نسبت به سایر گونه‌های حمله TMTO کاهش می‌یابد. افزون بر این، پیچیدگی زمان این روش نسبت به روش‌های دیگر TMTO به نسبت ۲ کمتر است. حمله Rainbow TMTO شامل دو مرحله اصلی با نام‌های مرحله پیش محاسبه و مرحله شکست است که در ادامه معرفی می‌گردند.

۲-۱- مرحله پیش محاسبه

اگر سیستم رمزنگاری دارای پیچیدگی فضای کلید N باشد، در حمله TMTO سعی می‌شود تمام جفت‌های کلید-کلمه رمز با رمز کردن کلمه ساده با N کلید ممکن تولید گردد. جفت کلید- کلمه رمز در زنجیرهایی سازمان‌دهی شده و به ازای هر زنجیر تنها عناصر ابتدایی و انتهایی در حافظه ذخیره می‌شوند. این امر سبب مصالحه یا معادلا کاهش حافظه در قبال افزایش زمان شکست (نسبت به حمله لغت‌نامه) می‌شود. زنجیرها به وسیله تابع کاهش R ساخته می‌شوند که کلید را از روی متن رمز شده قبلی می‌سازد. با به کار بردن متوالی تابع رمز S و تابع کاهش R، می‌توان زنجیرهایی شامل جفت‌های کلید- کلمه رمز متوالی ساخت:

$$K_i \xrightarrow{S_{k_i}(P_0)} C_i \xrightarrow{R(C_i)} K_{i+1} \qquad (۱)$$

که P_0 کلمه ساده ورودی و K_i و C_i به ترتیب کلید و متن رمز شده در iامین مرحله هستند. اگر توالی $R(S_{k_i}(P_0))$ را با $f(k)$ نمایش دهیم، زنجیره‌ای از کلیدها به‌صورت زیر داریم:

$$\dots \xrightarrow{f} K_i \xrightarrow{f} K_{i+1} \xrightarrow{f} K_{i+2} \xrightarrow{f} \dots \qquad (۲)$$

در حمله TMTO، m زنجیر با طول t تولید می‌شود و نقاط ابتدایی و انتهایی آن‌ها در حافظه ذخیره می‌شود. در روش حمله Rainbow، به‌جای استفاده از یک تابع کاهش واحد در هر جدول، t تابع کاهش متفاوت که به‌صورت شبه تصادفی تولید می‌شوند، به‌کار می‌روند تا یک زنجیر به طول t ساخته شود. بنابراین، زنجیرها در روش Rainbow به‌صورت زیر تولید می‌شوند:

$$\dots \xrightarrow{f_{i-1}} K_i \xrightarrow{fi} K_{i+1} \xrightarrow{f_{i+1}} K_{i+2} \xrightarrow{f_{i+2}} \dots \qquad (۳)$$

استفاده از توابع کاهش متفاوت در روش Rainbow، احتمال وقوع حوادث نامطلوب همانند ادغام و برخورد زنجیرها را نسبت به گونه‌هایی از حمله TMTO که از یک تابع کاهش ثابت استفاده می‌کنند، کاهش می‌دهد. البته وقوع حوادث نامطلوب با استفاده از توابع کاهش متفاوت، به‌طور کامل مرتفع نمی‌شود و به‌تبع آن احتمال موفقیت حمله تقریبا از رابطه زیر به‌دست آید [۲، ۸- ۷]:

$$P_{success} = 1 - \prod_{i=1}^{t}\left(1 - \frac{m_i}{N}\right),$$
$$m_1 = m, m_i = N\left(1 - \left(1 - \frac{1}{N}\right)^{m_{i-1}}\right) \approx N\left(1 - e^{-\frac{m_{i-1}}{N}}\right) \quad (۴)$$

در حمله Rainbow TMTO عملی، معمولا $N \gg m \gg 1$ می‌باشد، بنابراین:

$$m_i = N\left(1 - \left(1 - \frac{1}{N}\right)^{m_{i-1}}\right) \approx N\left(1 - \left(1 - \frac{m_{i-1}}{N}\right)\right) = m_{i-1} \Rightarrow m_i = m \quad (۵)$$

درنتیجه:

$$P_{success} \approx 1 - \left(1 - \frac{m}{N}\right)^{t} \approx 1 - e^{-\frac{mt}{N}} \quad (۶)$$

بر اساس رابطه (۶)، احتمال موفقیت برای مقادیر معین mt تقریبا ثابت است [۱۰]. به‌عبارت‌دیگر، برای یک احتمال موفقیت مشخص، مقدار mt باید تقریبا مقدار ثابتی باشد. به‌عنوان مثال، اگر پیچیدگی فضای کلید مورد جستجو N باشد و ما m زنجیر به طول t تولید کنیم به‌طوری‌که $mt = N$ باشد، احتمال موفقیت تقریبا $1 - e^{-1} = 0.63$ خواهد بود. به‌صورت کلی، برای احتمال موفقیت ثابت $P_{success}$، باید داشته باشیم:

$$mt \approx N \ln\left(\frac{1}{1 - P_{success}}\right) = kN, \quad k = \ln\left(\frac{1}{1 - P_{success}}\right) \quad (۷)$$

۲-۲- مرحله شکست

در این مرحله می‌توان کلید مربوط به متن رمز شده C را یافت مشروط بر آنکه این کلید در تولید زنجیری از جدول استفاده شده باشد. برای این منظور باید زنجیری با نقطه شروع $R(C)$ و طول t بسازیم. به این مرحله، بازسازی زنجیر (Chain Recovery) گفته می‌شود. هر عنصر از زنجیر بازیافت شده با تمام نقاط انتهایی مقایسه می‌گردد. در صورت تطابق با نقطه انتهایی یک زنجیر، کلیدی که دقیقا قبل از $R(C)$ در زنجیر مذکور قرار دارد کلید رمز است. ازآنجایی‌که برای هر زنجیر در جدول فقط نقاط ابتدایی و انتهایی آن ذخیره شده است، لازم است کلید مربوط به مرحله ماقبل $R(C)$ را دوباره از نقطه ابتدایی زنجیر بسازیم. می‌توان نشان داد که تعداد کل عملیات لازم برای بازسازی زنجیر در روش Rainbow برابر با $\frac{t(t-1)}{2} \approx \frac{t^2}{2}$ است. اگر هر عملیات T_{cpu} زمان نیاز داشته باشد، زمان لازم برای بازسازی زنجیر معادل $T_{cpu}\frac{t^2}{2}$ خواهد بود [۲، ۸- ۷].

هر عنصر زنجیر بازسازی‌شده می‌بایست با تمام نقاط پایانی ذخیره‌شده در جدول مقایسه شود تا زنجیرهایی که ممکن است کلید موردنظر در آن باشند، شناسایی گردند. اگر از جستجوی ترتیبی دیسک سخت با زمان دسترسی $T_{hardseq}$ برای مقایسه کلیدهای میانی استفاده کنیم، در بدترین حالت، زمان لازم برای مقایسه داده‌های زنجیر بازسازی‌شده با تمام نقاط پایانی، $m\,M_{ep}T_{hardseq}$ خواهد بود که M_{ep} میزان حافظه لازم برای ذخیره‌سازی نقطه پایانی هر زنجیر است. ازآنجایی‌که عملیات مقایسه در RAM صورت می‌گیرد، اطلاعات جدول لازم است از دیسک سخت به RAM انتقال یابد. اگر اطلاعات جدول مرتب شده باشد، با توجه به سرعت CPU و استفاده از جستجوی دودویی، زمان مقایسه در مقابل زمان انتقال اطلاعات از دیسک سخت به RAM ناچیز بوده و قابل صرف‌نظر کردن است. در جستجوی نشانه‌گذاری شده، به ازای هر المان زنجیر بازیافتی مقدار $\frac{m\,M_{ep}}{I}$ واحد از دیسک سخت فراخوانی می‌شود. در این رابطه I مبین تعداد نشانه (Index) است. اکنون اگر طول زنجیر t و زمان خواندن دیسک سخت در جستجوی نشانه‌گذاری‌شده $T_{hardInd}$ باشد، کل زمان لازم برای خواندن دیسک سخت برابر با $\frac{mt\,M_{ep}T_{hardInd}}{I}$ خواهد بود. دقت شود که فرض این است که نشانه‌ها در RAM ذخیره‌شده و یافتن نشانه در هر مرحله مقایسه زمان ناچیز دارد.

۳- ارائه مسئله بهینه‌سازی

پارامترهای حمله جدول Rainbow TMTO را می‌توان برای رسیدن به یک هدف مشخص بهینه کرد. در مسئله پیش رو هدف انتخاب پارامترهای حمله است به‌گونه‌ای که به ازای احتمال موفقیت مشخص و حداکثر زمان شکست معلوم، میزان حافظه موردنیاز کمینه شود. در ادامه، روند بهینه‌سازی مطلوب برای جستجوهای ترتیبی و نشانه‌گذاری‌شده در زیر بخش‌های مجزا انجام می‌شود.

۳-۱- تعریف مسئله برای جستجوی ترتیبی

در این قسمت، از جنبه عملی به حمله TMTO توجه می‌کنیم و می‌کوشیم میزان حافظه لازم جهت پیاده‌سازی حمله

را با توجه به حداکثر زمان شکست‌داده‌شده و احتمال موفقیت مشخص بهینه کنیم. بر اساس رابطه (۷)، اگر حاصل‌ضرب mt به‌گونه‌ای انتخاب شود که $mt = N\,ln\left(\frac{1}{1-P_{success}}\right) = kN$ ، آنگاه حصول احتمال موفقیت موردنظر با تقریب مناسبی تضمین می‌شود.

زمان شکست از دو قسمت مستقل تشکیل‌شده است، زمان بازسازی زنجیر و زمان جستجوی دیسک سخت. همان‌طور که قبلا نشان داده شد، زمان لازم برای بازسازی زنجیر $T_{cpu}\frac{t^2}{2}$ و زمان جستجوی دیسک سخت به‌صورت ترتیبی معادل با $m\,M_{ep}T_{hardseq}$ خواهد بود. با توجه به رابطه مستقیم میزان حافظه با تعداد زنجیرها m، مسئله بهینه‌سازی برای جستجوی ترتیبی را می‌توان به‌صورت زیر فرمول‌بندی کرد:

$$\min \quad m \qquad s.t. \begin{cases} mt = kN \\ \frac{T_{cpu}}{2}t^2 + m\,M_{ep}T_{hardseq} \le T_{online} \end{cases} \qquad (۸)$$

بر اساس محدودیت $mt = kN$، طول زنجیر t به m وابسته است. بنابراین:

$$\min \quad m \qquad \text{s.t.} \begin{cases} \frac{T_{cpu}}{2}(\frac{kN}{m})^2 + m\,M_{ep}T_{hardseq} \le T_{online} \end{cases} \qquad (۹)$$

یا به‌صورت معادل:

$$\min \quad m \qquad \text{s.t.} \begin{cases} m^3 - \frac{T_{online}}{M_{ep}T_{hardseq}}m^2 \\ +\frac{T_{cpu}}{2\,M_{ep}T_{hardseq}}k^2N^2 \le 0 \end{cases} \qquad (۱۰)$$

با تعریف:

$$x \triangleq m \ge 0, b \triangleq -\frac{T_{online}}{M_{ep}T_{hardseq}} \le 0, \quad c \triangleq \frac{T_{cpu}}{2\,M_{ep}T_{hardseq}}k^2N^2 \ge 0 \qquad (۱۱)$$

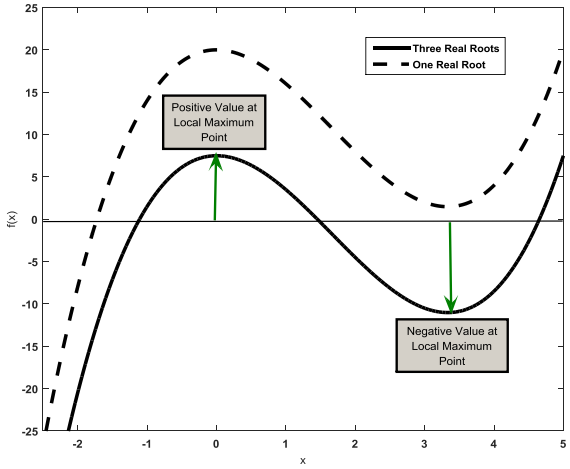
رابطه ساده زیر را برای بهینه‌سازی داریم:

$$\min \quad x \qquad \text{s.t.} \begin{cases} f(x) = x^3 + bx^2 + c \le 0, b \le 0, c \ge 0 \\ x \ge 0 \end{cases} \qquad (۱۲)$$

۳-۲- حل مسئله بهینه‌سازی برای جستجوی ترتیبی

شکل (۱) منحنی درجه سوم قید مسئله بهینه‌سازی تعریف‌شده در رابطه (۱۲) را در حالت نوعی نشان می‌دهد.

منحنی قید دارای یک بیشینه محلی مثبت در مبدأ و یک کمینه محلی منفی در سمت راست مبدأ است. برای حل مسئله بهینه‌سازی، باید کوچک‌ترین مقدار مثبت x را بیابیم به‌گونه‌ای که مقدار منحنی به ازای آن منفی یا صفر شود.



شکل (۱). شمای نوعی منحنی درجه سوم قید مسئله بهینه‌سازی

واضح است که اگر مقدار کمینه محلی مثبت باشد، منحنی قید با سمت راست محور xها تلاقی نداشته و مسئله بهینه‌سازی جواب حقیقی نخواهد داشت. در نتیجه، برای داشتن جواب حقیقی مقدار کمینه محلی می‌بایست منفی باشد:

$$f(\text{local min}) \le 0 \Rightarrow \Delta \triangleq 4b^3 + 27c \le 0 \qquad (۱۳)$$

برای داشتن جواب حقیقی، باید نامساوی (۱۳) برقرار باشد. اگر $\Delta \le 0$ باشد، منحنی قید سه برخورد با محور xها خواهد داشت که دو تای آن‌ها مثبت و یکی منفی است. به وضوح کوچک‌ترین مقدار مثبت که منحنی قید را صفر یا منفی می‌کند همان کوچک‌ترین تقاطع مثبت منحنی قید با محور xها و یا به عبارت دیگر، کوچک‌ترین ریشه مثبت معادله $x^3 + bx^2 + c = 0$ می‌باشد:

$$x_{opt} = \min_{positive\ value}\{x_1, x_2, x_3\} \qquad (۱۴)$$

که در آن، x_1, x_2, x_3 ریشه‌های معادله $x^3 + bx^2 + c = 0$ می‌باشد. بر اساس فرمول‌های کاردانو، مقدار ریشه‌ها برحسب ضرایب b و c از روابط ذیل به‌دست می‌آیند [۱۲-۱۱]:

$$x_1 = \frac{1}{3}\left\{\sqrt[3]{\frac{\sqrt{27c\Delta} - \Delta + 2b^3}{2}} + \sqrt[3]{\frac{2b^6}{\sqrt{27c\Delta} - \Delta + 2b^3}} - b\right\} \qquad (۱۵)$$

$$x_2 = \frac{1}{3}\{-\frac{1-j\sqrt{3}}{2}\sqrt[3]{\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}}-\frac{1+j\sqrt{3}}{2}\sqrt[3]{\frac{2b^6}{\sqrt{27c\Delta}-\Delta+2b^3}}-b\} \quad (۱۶)$$

$$x_3 = \frac{1}{3}\{-\frac{1+j\sqrt{3}}{2}\sqrt[3]{\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}}-\frac{1-j\sqrt{3}}{2}\sqrt[3]{\frac{2b^6}{\sqrt{27c\Delta}-\Delta+2b^3}}-b\} \quad (۱۷)$$

اکنون مسئله بهینه‌سازی به انتخاب کوچک‌ترین ریشه مثبت x_1, x_2, x_3 تبدیل می‌شود. ازآنجایی‌که $\Delta \le 0$ است، $\sqrt{27c\Delta}$ مقداری موهومی است و درنتیجه:

$$\sqrt[3]{\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}} = \alpha + j\beta = re^{j\theta} \quad (۱۸)$$

که در آن:

$$\begin{aligned}\alpha + j\beta &= \sqrt[3]{\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}} \\ &= \left(\frac{j\sqrt{27c|\Delta|}-\Delta+2b^3}{2}\right)^{\frac{1}{3}} \\ &= \left(\frac{27c|\Delta|+(2b^3-\Delta)^2}{4}\right)^{\frac{1}{6}}\cos\left(\frac{1}{3}\psi\right) \\ &+ j\left(\frac{27c|\Delta|+(2b^3-\Delta)^2}{4}\right)^{\frac{1}{6}}\sin(\frac{1}{3}\psi) \\ &= r\cos(\theta) + j\, r\sin(\theta)\end{aligned} \quad (۱۹)$$

و:

$$\psi = \begin{cases} tan^{-1}(\frac{\sqrt{27c|\Delta|}}{2b^3-\Delta}), & 2b^3-\Delta \ge 0 \\ \pi + tan^{-1}(\frac{\sqrt{27c|\Delta|}}{2b^3-\Delta}), & 2b^3-\Delta < 0 \end{cases} \quad (۲۰)$$

به‌علاوه می‌توان ثابت کرد که:

$$r = |b| \Rightarrow b^2 = r^2 = \alpha^2 + \beta^2 \quad (۲۱)$$

با توجه به $\alpha, \beta \in \mathbb{R}$ و با استفاده از اتحاد (۲۱)، ریشه‌ها را می‌توان به شکل دیگر نوشت. به‌عنوان مثال برای x_1:

$$x_1 = Re\{x_1\} + jIm\{x_1\} = \frac{2\alpha - b}{3} \in \mathbb{R} \quad (۲۲)$$

و به‌طور مشابه:

$$x_2 = Re\{x_2\} + jIm\{x_2\} = \frac{-\alpha - b - \sqrt{3}\beta}{3} \in \mathbb{R} \quad (۲۳)$$

$$x_3 = Re\{x_3\} + jIm\{x_3\} = \frac{-\alpha - b + \sqrt{3}\beta}{3} \in \mathbb{R} \quad (۲۴)$$

این روابط جدید نشان می‌دهند که ریشه‌ها حقیقی هستند و این مسئله با شرط اولیه $\Delta \le 0$ همخوانی دارد. با مراجعه به رابطه (۱۸)، $\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}$ سومین ریشه $\alpha + j\beta = re^{j\theta}$ است که عددی مختلط با قسمت موهومی مثبت است. بنابراین، $\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}$ در ربع اول یا دوم صفحه اعداد مختلط واقع بوده و درنتیجه ریشه سوم آن یعنی $\alpha + j\beta = re^{j\theta}$ بالاجبار در ربع اول واقع است. پس:

$$\alpha, \beta \ge 0 \quad (۲۵)$$

$$0 \le \theta \le \frac{\pi}{3} \Rightarrow 0 \le \frac{\beta}{\alpha} = \tan(\theta) \le \sqrt{3} \quad (۲۶)$$

حال نشان خواهیم داد:

$$x_2 \le x_3 \le x_1 \quad (۲۷)$$

نخست فرض کنیم شرط $x_2 \le x_3$ درست باشد. بنابراین:

$$x_2 \le x_3 \Leftrightarrow \frac{-\alpha - b - \sqrt{3}\beta}{3} \le \frac{-\alpha - b + \sqrt{3}\beta}{3} \Leftrightarrow -\beta \le \beta \Leftrightarrow \beta \ge 0 \quad (۲۸)$$

با توجه به رابطه (۲۵) و برگشت‌پذیر بودن روابط در (۲۸)، عبارت $x_2 \le x_3$ درست است. با تکرار همین روش برای $x_3 \le x_1$ خواهیم داشت:

$$x_3 \le x_1 \Leftrightarrow \frac{-\alpha - b + \sqrt{3}\beta}{3} \le \frac{2\alpha - b}{3} \Leftrightarrow \sqrt{3}\beta \le 3\alpha \Leftrightarrow \frac{\beta}{\alpha} = \tan(\theta) \le \sqrt{3} \quad (۲۹)$$

برای $\Delta \le 0$ دو ریشه مثبت و یک ریشه منفی داریم. با توجه به رابطه (۲۷)، تنها حالت زیر ممکن خواهد بود:

$$x_2 \le 0 \le x_3 \le x_1 \quad (۳۰)$$

با مراجعه به روابط (۱۴) و (۳۰) خواهیم داشت:

$$x_{opt} = \min_{positive\ value}\{x_1, x_2, x_3\} = \min_{positive\ value}\{x_2 \le 0 \le x_3 \le x_1\} = x_3 \quad (۳۱)$$

و درنهایت:

$$m_{opt} = x_{opt} = x_3 = \frac{1}{3}\left\{-\frac{1+j\sqrt{3}}{2}\sqrt[3]{\frac{\sqrt{27c\Delta}-\Delta+2b^3}{2}} - \frac{1-j\sqrt{3}}{2}\sqrt[3]{\frac{2b^6}{\sqrt{27c\Delta}-\Delta+2b^3}} - b\right\} = \frac{-\alpha-b+\sqrt{3}\beta}{3} \quad (۳۲)$$

۳-۳- مقادیر بهینه پارامترها و حداقل حداکثر زمان شکست برای جستجوی ترتیبی

با استفاده از روابط (۷) و (۳۲)، مقدار بهینه برای طول زنجیرها و میزان حافظه موردنیاز عبارت‌اند از:

$$Mem_{opt} = m_{opt}(M_{sp} + M_{ep}) \quad (۳۳)$$

$$Mem_{opt} = m_{opt}(M_{sp} + M_{ep}) \quad (۳۴)$$

که $m_{opt} = x$ تعداد بهینه زنجیرهاست که با جایگذاری مناسب از رابطه (۳۲) به دست می‌آید. M_{sp} و M_{ep} به ترتیب میزان حافظه موردنیاز برای ذخیره کردن نقاط ابتدایی و انتهایی هر زنجیر می‌باشد. برای اینکه مسئله بهینه‌سازی جواب داشته باشد می‌بایست $\Delta \le 0$ باشد بنابراین، حداقل مقدار برای حداکثر زمان شکست $T_{feasible}$ مربوط به شرایطی است که $\Delta = 0$ باشد:

$$\Delta = 0 \Rightarrow 4b^3 + 27c = 0 \quad (۳۵)$$

با جایگذاری b و c در رابطه (۳۵) خواهیم داشت:

$$T_{feasible} = \frac{3}{2}\left(M_{ep}T_{hardseq}N\sqrt{T_{cpu}}\ln\left(\frac{1}{1-P_{success}}\right)\right)^{2/3} \quad (۳۶)$$

اگر T_{online} داده‌شده از $T_{feasible}$ کمتر باشد، مسئله بهینه‌سازی جواب حقیقی نخواهد داشت.

۴-۳- تعریف مسئله برای جستجوی نشانه‌گذاری شده

در این قسمت، تحلیل مسئله پیشین را برای جستجوی نشانه‌گذاری شده پی می‌گیریم. به‌طور مشابه برای حصول احتمال موفقیت معین باید رابطه $mt = N\ln\left(\frac{1}{1-P_{success}}\right) = kN$ برقرار باشد. مجددا زمان شکست از دو قسمت مستقل تشکیل‌شده است، زمان بازسازی زنجیر و زمان جستجوی دیسک سخت. همان‌طور که قبلا نشان داده شد، زمان لازم برای بازسازی زنجیر $T_{cpu}\frac{t^2}{2}$ و زمان جستجوی دیسک سخت به‌صورت نشانه‌گذاری شده معادل با $\frac{mt\,M_{ep}T_{hardInd}}{I}$ خواهد بود. نهایتا مسئله بهینه‌سازی برای جستجوی نشانه‌گذاری شده به شیوه زیر قابل بیان است:

$$\min\ m \quad \text{s.t.}\ \begin{cases} mt = kN \\ \frac{T_{cpu}}{2}t^2 + \frac{mt\,M_{ep}T_{hardInd}}{I} \le T_{online} \end{cases} \quad (۳۷)$$

بر مبنای رابطه (۷)، طول زنجیر t به m وابسته است. بنابراین:

$$\min\ \frac{kN}{t} \quad \text{s.t.}\ \left\{\frac{T_{cpu}}{2}t^2 + \frac{kN\,M_{ep}T_{hardInd}}{I} \le T_{online}\right. \quad (۳۸)$$

یا به‌صورت معادل:

$$\max\ t \quad \text{s.t.}\ \begin{cases} t^2 \le \dfrac{T_{online} - \dfrac{kN\,M_{ep}T_{hardInd}}{I}}{\dfrac{T_{cpu}}{2}} \end{cases} \quad (۳۹)$$

با تعریف:

$$y \overset{\text{def}}{=} t \ge 0, d \overset{\text{def}}{=} \frac{T_{online} - \frac{kN\,M_{ep}T_{hardInd}}{I}}{\frac{T_{cpu}}{2}} \quad (۴۰)$$

رابطه ساده زیر را برای بهینه‌سازی داریم:

$$\max\ y \quad \text{s.t.}\ \{y^2 \le d \quad (۴۱)$$

۵-۳- حل مسئله بهینه‌سازی برای جستجوی نشانه‌گذاری‌شده

حل مسئله بهینه‌سازی در حالت جستجوی نشانه‌گذاری‌شده ساده است. ابتدا باید توجه داشت برای حصول جواب حقیقی باید $d \ge 0$. در صورت برقراری شرط به وضوح جواب برابر است با:

$$y_{opt} = \sqrt{d} \quad (۴۲)$$

۶-۳-مقادیر بهینه پارامترها و حداکثر زمان شکست برای جستجوی نشانه‌گذاری‌شده

با استفاده از روابط (۷) و (۴۲)، مقدار بهینه برای طول زنجیرها و میزان حافظه مورد نیاز عبارت‌اند از:

$$Mem_{opt} = m_{opt}(M_{sp} + M_{ep}) \quad (۴۳)$$

$$Mem_{opt} = m_{opt}(M_{sp} + M_{ep}) \quad (۴۴)$$

که $t_{opt} = y$ تعداد بهینه زنجیرهاست که با جایگذاری در رابطه (۴۲) به‌دست می‌آید. M_{sp} و M_{ep} به ترتیب میزان حافظه موردنیاز برای ذخیره کردن نقاط ابتدایی و انتهایی هر زنجیر می‌باشد. برای اینکه مسئله بهینه‌سازی جواب داشته باشد می‌بایست $d \geq 0$ باشد بنابراین، کم‌ترین مقدار برای حداکثر زمان شکست $T_{feasible}$ مربوط به شرایطی است که $d = 0$ باشد:

$$T_{feasible} = \frac{kN\, M_{ep} T_{hardInd}}{I} \quad (۴۵)$$

اگر T_{online} داده‌شده از $T_{feasible}$ کمتر باشد، مسئله بهینه‌سازی جواب حقیقی نخواهد داشت.

۴- مثال عملی

به‌عنوان یک مثال عملی حمله Rainbow TMTO به الگوریتم رمز DES را مدنظر قرار می‌دهیم [۱۳-۱۴]. پیچیدگی فضای جستجو در الگوریتم DES برابر با $N = 2^{56}$ است. طول کلید و کلمات در DES، ۶۴ بیت است منتها ۸ بیت در کلید به‌عنوان توازن در نظر گرفته می‌شود و ازاین‌رو پیچیدگی فضای کلید $N = 2^{56}$ است. با توجه به طول ۶۴ بیتی کلمات $M_{sp} = M_{ep} = 8\ B$. فرض می‌کنیم مرحله شکست توسط یک رایانه معمولی با پردازنده Core i7 و دیسک سخت 7200 rpm SATA-3 انجام می‌شود. ارزیابی‌های عملی نشان داد که زمان دسترسی ترتیبی در دیسک سخت تقریبا $T_{hardseq} = 3.33 \times 10^{-9} \frac{s}{B}$، زمان دسترسی نشانه‌گذاری‌شده به دیسک سخت تقریبا $T_{hardInd} = 1.05 \times 10^{-7} \frac{s}{B}$ و زمان موردنیاز برای انجام یک الگوریتم کامل DES حدودا $T_{cpu} = 5.92 \times 10^{-9} \frac{s}{op}$ می‌باشد. درنهایت فرض می‌کنیم احتمال موفقیت $P_{success} = 0.86$ برای حمله کفایت می کند.

بر اساس رابطه (۳۶)، کمینه مقدار حداکثر زمان شکست قابل دسترسی در جستجوی ترتیبی عبارتست از:

$$T_{online} \geq T_{feasible} = \frac{3}{2}\left(8 \times\ 3.33 \times 10^{-9} \times 2^{56} \times \sqrt{5.92 \times 10^{-9} \times 2}\right)^{\frac{2}{3}} = 6653\ s = 1.85\ h \quad (۴۶)$$

با در نظر گرفتن زمان شکست $T_{online} = 2.00\ h$ ، مقادیر b و c تعریف شده در رابطه (۱۱) به صورت زیر محاسبه می‌شود:

$$b \overset{\text{def}}{=} -\frac{T_{online}}{M_{ep} T_{hardseq}} = -\frac{7200}{8 \times\ 3.33 \times 10^{-9}} = -2.71 \times 10^{11} \quad (۴۷)$$

$$c \overset{\text{def}}{=} \frac{T_{cpu}}{2\, M_{ep} T_{hardseq}} k^2 N^2 = \frac{5.92 \times 10^{-9} \times 2^2 \times 2^{112}}{2 \times 8 \times 3.33 \times 10^{-9}} = 2.31 \times 10^{33} \quad (۴۸)$$

درنهایت، با استفاده از روابط (۳۳) و (۳۴) مقادیر بهینه طول زنجیر و حافظه موردنیاز برای جستجوی ترتیبی به دست می‌آیند:

$$m_{opt} = 1.26 \times 10^{11} \approx 2^{37} \quad (۴۹)$$

$$Mem = m_{opt}(M_{sp} + M_{ep}) = 16 \times 1.26 \times 10^{11} = 2.02 \times 10^{12} \approx 2^{41} B = 2\ TB \quad (۵۰)$$

$$t_{opt} = \frac{kN}{m_{opt}} = 1.14 \times 10^{6} \approx 2^{20} \quad (۵۱)$$

لازم به ذکر است که برای ذخیره کردن نقاط ابتدایی و انتهایی نیاز به ۱۶ بایت حافظه است بنابراین $(M_{sp} + M_{ep})$ در رابطه (۵۰) برابر ۱۶ در نظر گرفته شده است. همین روند را می توان برای جستجوی نشانه‌گذاری‌شده به‌کار برد. با فرض 3 بایت برای نشانه، $I = 2^{24}$ خواهد بود. درنتیجه:

$$T_{online} \geq T_{feasible} = \frac{kN\, M_{ep} T_{hardInd}}{I} = \frac{2 \times 2^{56} \times 8 \times 1.05 \times 10^{-7}}{2^{24}} = 7216s = 2.00\ h \quad (۵۲)$$

با در نظر گرفتن زمان شکست $T_{online} = 2.10\ h$، مقادیر بهینه حافظه موردنیاز و طول زنجیر بر اساس روابط داده‌شده به‌دست می‌آیند:

$$d \overset{\text{def}}{=} \frac{T_{online} - \frac{kN\, M_{ep} T_{hardInd}}{I}}{\frac{T_{cpu}}{2}} = 1.22 \times 10^{11} \quad (۵۳)$$

$$t_{opt} = \sqrt{d} = 3.49 \times 10^{5} \approx 2^{18} \quad (۵۴)$$

$$Mem = \frac{kN}{t_{opt}}\left(M_{sp} + M_{ep}\right) = 16 \times 4.13 \times 10^{11} = 6.61 \times 10^{12} \approx 2^{43} B = 8\ TB \quad (۵۵)$$

به عنوان مثال دیگر، فرض کنید که قصد شکست یک کلمه رمز ۸ حرفی Hash شده توسط الگوریتم SHA1 را داریم [۱۳]، [۱۴]. هریک از حروف کلمه رمز از مجموعه ۹۵ حالتی شامل حروف، اعداد و کاراکترهای خاص یک صفحه‌کلید انتخاب می‌شود. درنتیجه پیچیدگی فضای رمز برابر با $N = 95^8$ خواهد بود. طول Hash در الگوریتم SHA1 برابر ۲۰ بایت است بنابراین $M_{ep} = 20\ B$ است. عملیات شکست را مجددا توسط رایانه با

پردازنده Core i7 و دیسک سخت ۷۲۰۰ rpm SATA-3 انجام می‌دهیم. در این رایانه زمان دسترسی ترتیبی در دیسک سخت تقریبا $T_{hardseq} = 3.33 \times 10^{-9} \frac{s}{B}$ ، زمان دسترسی نشانه‌گذاری شده به دیسک سخت تقریبا $T_{hardind} = 1.05 \times 10^{-7} \frac{s}{B}$ و زمان موردنیاز برای انجام یک الگوریتم کامل SHA1 حدودا $T_{cpu} = 0.85 \times 10^{-9} \frac{s}{op}$ می‌باشد. مجددا با در نظر گرفتن احتمال موفقیت $P_{success} = 0.86$ و بر مبنای رابطه (۳۶)، کمینه مقدار حداکثر زمان شکست قابل دسترسی در جستجوی ترتیبی عبارتست از:

$$T_{online} \geq T_{feasible} = \frac{3}{2}\left(20 \times 3.33 \times 10^{-9} \times 95^8 \times \sqrt{0.85 \times 10^{-9} \times 2}\right)^{\frac{2}{3}} = 1309\ s = 0.36\ h \quad (۵۶)$$

با در نظر گرفتن زمان شکست $T_{online} = 1\ h$ ، مقادیر b و c تعریف‌شده در رابطه (۱۱) به‌صورت زیر محاسبه می‌شود:

$$b \stackrel{\text{def}}{=} -\frac{T_{online}}{M_{ep} T_{hardseq}} = -\frac{3600}{20 \times 3.33 \times 10^{-9}} = -5.41 \times 10^{10} \quad (۵۷)$$

$$c \stackrel{\text{def}}{=} \frac{T_{cpu}}{2\, M_{ep} T_{hardseq}} k^2 N^2 = \frac{0.85 \times 10^{-9} \times 2^2 \times 95^{16}}{2 \times 20 \times 3.33 \times 10^{-9}} = 1.12 \times 10^{30} \quad (۵۸)$$

درنهایت، با استفاده از روابط (۳۳) و (۳۴) مقادیر بهینه طول زنجیر و حافظه موردنیاز برای جستجوی ترتیبی به‌دست می‌آیند:

$$m_{opt} = 4.76 \times 10^9 \approx 2^{32} \quad (۵۹)$$

$$Mem = m_{opt}\left(M_{sp} + M_{ep}\right) = 28 \times 4.76 \times 10^9 = 1.33 \times 10^{11} \approx 2^{37} B = 128\ \mathrm{GB} \quad (۶۰)$$

$$t_{opt} = \frac{kN}{m_{opt}} = 2.79 \times 10^6 \approx 2^{21} \quad (۶۱)$$

لازم به ذکر است که برای ذخیره کردن نقاط ابتدایی و انتهایی نیاز به 28 بایت حافظه است (8 بایت برای کلمه رمز و 20 بایت برای Hash) بنابراین $\left(M_{sp} + M_{ep}\right)$ در رابطه (۶۰) برابر 28 در نظر گرفته‌شده است. همین روند را می‌توان برای جستجوی نشانه‌گذاری شده به‌کار برد. با فرض ۳ بایت برای نشانه، $I = 2^{24}$ خواهد بود. درنتیجه:

$$T_{online} \geq T_{feasible} = \frac{kN\, M_{ep} T_{hardInd}}{I} = \frac{2 \times 95^8 \times 20 \times 1.05 \times 10^{-7}}{2^{24}} = 1661\ s = 0.46\ h \quad (۶۲)$$

با در نظر گرفتن زمان شکست $T_{online} = 1\ h$، مقادیر بهینه حافظه موردنیاز و طول زنجیر بر اساس روابط داده‌شده به‌دست می‌آیند:

$$d \stackrel{\text{def}}{=} \frac{T_{online} - \frac{kN\, M_{ep} T_{hardInd}}{I}}{\frac{T_{cpu}}{2}} = 4.57 \times 10^{12} \quad (۶۳)$$

$$t_{opt} = \sqrt{d} = 2.14 \times 10^6 \approx 2^{21} \quad (۶۴)$$

$$Mem = \frac{kN}{t_{opt}}\left(M_{sp} + M_{ep}\right) = 28 \times 6.20 \times 10^9 = 1.74 \times 10^{11} \approx 2^{37} B = 128\ GB \quad (۶۵)$$

۵- نتیجه‌گیری

در این مقاله با فرض دو شیوه جستجوی ترتیبی و نشانه‌گذاری‌شده دیسک سخت در مرحله شکست، روش انتخاب بهینه پارامترهای حمله Rainbow TMTO ارائه شد. برخلاف کارهای پیشین که عمدتا بهینه‌سازی بدون در نظر گرفتن زمان شکست انجام شده است، در این نوشتار به زمان شکست به‌طور ویژه توجه شده است. هدف از بهینه‌سازی ارائه‌شده، حصول کمینه مقدار حافظه موردنیاز مشروط بر حداکثر زمان شکست معین و احتمال شکست مشخص بود. در برخی موارد حل مسائل بهینه‌سازی تعریف‌شده نیازمند استفاده از روابط نسبتا پیچیده‌ای همچون روابط کاردانو بود. صرف نظر از تحلیل نسبتا پیچیده مسائل، به‌عنوان نتیجه، مجموعه‌ای از روابط ساده برای محاسبه مقادیر پارامترهای حمله Rainbow TMTO برای جستجوی ترتیبی و نشانه‌گذاری‌شده ارائه شد. نهایتا شیوه استفاده از روابط با ارائه چند مثال نمونه تصدیق شد.

۶- مراجع

[1] M. E. Hellman, "A Cryptanalytic Time-Memory Trade-off", IEEE Transactions on Information Theory, pages 401–406, 1980.

[2] P. Oechslin, "Making a Faster Cryptanalytic Time-Memory Trade-of", volume 2729 of Lecture Notes in Computer Science, pages 617–630, Springer, 2003.

[3] I. J. Kim and T. Matsumoto, "Achieving Higher Success Probability in Time-Memory Trade-off Cryptanalysis without Increasing Memory Size", IEICE Transactions on Fundamentals, pages 123–129, 1999.

[4] NurdanSaran, Ali Doganaksoy, "Choosing Parameters to Achieve A Higher Success Rate for Hellman Time Memory Trade Off Attack", International Conference on Availability, Reliability and Security, Mar. 2009.

[5] K. Kusuda and T. Matsumoto, "Optimization of Time-Memory Trade-off Cryptanalysis and Its Application to DES", IEICE Transactions on Fundamentals, pages 35–48, 1996.
[6] Vrizlynn L. L. Thing, Hwei-Ming Ying, "Rainbow Table Optimization for Password Recovery", International Journal on Advances in Software, vol 4 no 3 & 4, 2011.
[7] G. Avoine, P. Junod, and P. Oechslin, "Characterization and Improvement of Time-Memory Trade-off Based on Perfect Tables", ACM Transactions on Information Systems, 2008.
[8] G. Avoine, P. Junod, and P. Oechslin, "Time-Memory Trade-offs: False Alarm Detection using Checkpoints", INDOCRYPT, volume 3797 of Lecture Notes in Computer Science, pages 183–196, Springer, 2005.
[9] F. Broek and E. Poll, "A Comparison of Time-Memory Trade-Off Attacks on Stream Ciphers", AFRICACRYPT 2013, pp. 406–423, 2013.
[10] J. Hong, S. Moon, "A comparison of cryptanalytic tradeoff algorithms", Cryptology ePrint Archive. Report 2010/176.
[11] Murray R. Spiegel, Seymour Lipschutz, John Liu, "Mathematical Handbook of Formulas and Tables", McGraw-Hill, 1968.
[12] L. G. Dickson, "Elementary Theory of Equations", John Wiley and Sons Incorporation, 1914.
[13] William Stallings, "Cryptography and Network Security Principles and Practices", 4th Ed., Prentice Hall, Nov. 16, 2005.
[14] D. R. Stinson, "Cryptography Theory and Practice", 3rd Ed., Chapman & Hall/CRC, 2006.

Journal Of Electronical & Cyber Defence
Vol. 3, No. 3, 2016, Serial No.11

Optimum Parameter Selection for Rainbow Table TMTO Attack Considering Breaking Time and Using Sequential and Index Search Methods

M. Hadi*, M. Moeini Jahromi

PhD student at Sharif University of Technology

(Received: 02/02/2015, Accepted: 12/01/2016)

ABSTRACT

Required memory, online search time and success probability are the main performance metrics of a Time Memory Trade-Off (TMTO) Attack. One of the basic challenges in TMTO attack is the way of choosing TMTO attack parameters like number and length of chains to meet some certain performance metrics. Considering online breaking time, we propose an optimized procedure for selecting rainbow table TMTO attack parameters. Unlike previous works that mainly deal with minimizing required memory in the rainbow table TMTO attack, we simultaneously focus on the required memory and the online breaking time and consider index and sequential search techniques. Our parameter selection technique is optimized regarding the minimization of the required memory subject to a certain success probability and a maximum online breaking time. Obtained results are two compact mathematical expressions for determining the rainbow table TMTO attack parameters, number and length of chains for the sequential and the index search methods. The application of our optimized parameter selection procedure is also shown in few sample design examples.

Keywords: Time-Memory Trade-Off (TMTO) Attack, Breaking Time, Sequential Search, Index Search, Constrained Optimization

.

* Corresponding Author Email: mhadi@ee.sharif.edu