امنیت ساختارهای اسفنجی

اکرم خالصی و محمدعلی ارومیه‌چی‌ها*

پژوهش‌گر پژوهشگاه توسعه فناوری‌های پیشرفته، پژوهشکده افتا،
گروه رمز و امنیت اطلاعات، تهران، ایران
khalesiakram@yahoo.com
Orumiehchiha@rcdat.ir

چکیده

ساختار اسفنجی، ساختاری پرکاربرد در طراحی الگوریتم‌های رمزنگاری است که طراحی الگوریتم را به طراحی یک جایگشت یا تبدیل شبه‌تصادفی کاهش می‌دهد. گسترش الگوریتم‌های مبتنی بر ساختار اسفنجی و انتخاب طرح‌های مبتنی بر این ساختار در مسابقات SHA3 و CAESAR ضرورت بررسی امنیت آن را در برابر انواع حملات افزایش می‌دهد. در این نوشتار با محوریت امنیت ساختارهای اسفنجی، به مطالعه روش‌های تحلیل عام روی این ساختار می‌پردازیم و پیچیدگی آنها را بررسی می‌کنیم. درنظرگرفتن پیچیدگی‌های معرفی‌شده برای حملات عام در انتخاب پارامترهای ساختار اسفنجی در زمان طراحی الگوریتم، برای رسیدن به یک سطح امنیتی مشخص، ضروری بوده و به همین سبب مقاله حاضر هم از حیث طراحی الگوریتم‌های مبتنی بر اسفنج و هم از دید تحلیل این الگوریتم‌ها حائز اهمیت است. پیشنهاد می‌شود مقاله "ساختار اسفنجی؛ معرفی و کاربردها" را که در همین نشریه به چاپ رسیده است قبل از مطالعه این مقاله بررسی و مرور شود.

واژگان کلیدی: ساختار اسفنج، الگوریتم‌های مبتنی بر اسفنج، برخورد، پیش تصویر، پیش تصویر دوم، بازیابی حالت داخلی، انقیاد خروجی

۱- مقدمه

ساختار اسفنجی[1] [۱] ابزاری پرکاربرد در رمزنگاری است که می‌تواند در طراحی طیف وسیعی از الگوریتم‌ها شامل رمزهای دنباله‌ای، توابع چکیده‌ساز [۲، ۳، ۴]، مولد اعداد شبه‌تصادفی با امکان بارگذاری مجدد هسته [۵]، توابع تولید کلید، کدهای احراز اصالت پیام و رمزنگاری احراز اصالت‌شده [۶] مورد استفاده قرار گیرد.

ایده ساختار اسفنجی در زمان طراحی تابع چکیده‌ساز RADIOGATUN شکل گرفته است [۷]. برگزیده‌شدن الگوریتم تابع چکیده‌ساز Keccak با ساختار اسفنجی در مسابقه SHA3 [۸] و الگوریتم رمزنگاری احراز اصالت‌شده ASCON با ساختار اسفنجی در مسابقه CAESAR [۹] گواهی بر قابلیت‌های این ساختار در رمزنگاری است.

الگوریتم Keccak چکیده‌سازی مبتنی بر ساختار اسفنجی، برگزیده مسابقه SHA3 و آخرین عضو اضافه‌شده به FIPS 180-4 (استاندارد مربوط به توابع چکیده‌ساز) است[۱۰]. طراحان Keccak با استفاده از جایگشت به‌کاررفته در این طرح، الگوریتم‌های دیگری شامل الگوریتم‌های رمزنگاری احراز اصالت‌شده Ketje و Keyak و تابع چکیده‌ساز پرسرعت Kangaroo Twelve را نیز ارائه کرده‌اند [۱۱].

ساختار اسفنجی امکان تطبیق الگوریتم با پارامترهای مختلف برای امنیت، سرعت و منابع را تأمین می‌کند؛ هم‌چنین که الگوریتم ASCON الگوریتم رمزنگاری احراز اصالت‌شده پیشنهادی مؤسسه NIST[2] در کاربردهای رمزنگاری سبک وزن[3] و Keccak خانواده‌ای از توابع چکیده‌ساز، قابل تنظیم با توجه به نیازمندی‌ها در کاربری‌های مختلف است.

از ویژگی‌های مثبت ساختار اسفنجی امکان ارائه امنیت اثبات‌پذیر مشابه یک مد عملکرد برای رمز قالبی است. بدین ترتیب، ارزیابی امنیت الگوریتمی مبتنی بر این ساختار به ارزیابی امنیت یک تبدیل و یا یک جایگشت تصادفی کاهش می‌یابد. این ویژگی قابلیت اعتماد به طرحی را که مبتنی بر این ساختار باشد، افزایش خواهد داد.

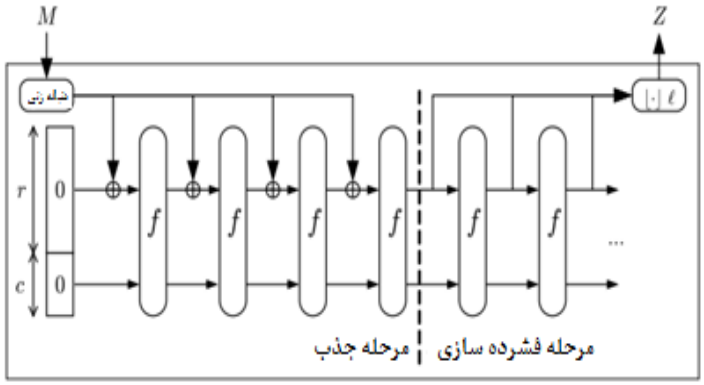
حملات روی الگوریتم‌های مبتنی بر ساختار اسفنجی را می‌توان به دو دسته تقسیم کرد: حملات مبتنی بر تابع به‌روزرسانی حالت داخلی اسفنج [۱۲، ۱۳، ۱۴] و حملات مستقل از تابع به‌روزرسانی [۱].

[1] Sponge structure

[2] National Institute of Statndards and Technology
[3] Light-weight cryptography

* نویسندهٔ عهده‌دار مکاتبات

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]



(شکل-۱): ساختار اسفنجی $Z = SPONGE[f, pad, r](M, l)$ [۱]

در حالت کلی، اگر یک حمله روی ساختار اسفنجی از خصوصیات تبدیل یا جایگشت مورد استفاده در ساختار استفاده نکند، حمله‌ای عمومی خواهد بود. در این نوشتار به مطالعه حملات عام روی ساختار اسفنجی پرداخته و نحوه اعمال مؤثر این حملات را به همراه پیچیدگی آنها با استفاده از نمایش گرافیکی برای ساختار اسفنجی بیان می‌کنیم. با مشخص‌شدن پیچیدگی این حملات، تعیین پارامترهای ساختار اسفنجی برای رسیدن به یک سطح امنیتی مشخص به‌سادگی میسر خواهد شد.

در بخش دوم این نوشتار به بیان تعاریف مورد نیاز پرداخته و در بخش‌های بعد به‌ترتیب حملات عام برخورد داخلی، یافتن مسیر به یک حالت داخلی، تشخیص دور در خروجی، بازیابی حالت داخلی و انقیاد خروجی[1] را مورد مطالعه قرار می‌دهیم.

۲- تعاریف

در این بخش تعاریف، اصطلاحات و توابع کمکی را در بحث ساختارهای اسفنجی که در بخش‌های بعدی استفاده می‌شوند، معرفی می‌کنیم.

۲-۱- رشته‌بیت

طول رشته‌بیت M بر حسب بیت را با $|M|$ نمایش می‌دهیم. رشته‌بیت M می‌تواند به‌عنوان دنباله‌ای از قالب‌ها با طول ثابت x در نظر گرفته شود که در آن طول قالب آخر ممکن است کوتاه‌تر باشد. تعداد قالب‌های M با $|M|_x$ نمایش داده می‌شود. قالب‌های M با M_i نمایش داده شده و اندیس آن از صفر تا $|M|_x - 1$ تغییر می‌کند. طول یک رشته تهی برابر صفر بوده و بیتی ندارد. رشته تهی را می‌توان به‌صورت دنباله‌ای بدون قالب و یا یک قالب با طول صفر در نظر گرفت. در این مستند، درصورتی‌که اشاره صریح نشده باشد، رشته تهی را بدون قالب در نظر می‌گیریم.

تفکیک رشته بیت M به l بیت ابتدایی آن را با $\lfloor M \rfloor_l$ نمایش می‌دهیم. یک رشته بیت متشکل از n صفر را با 0^n و پیوست[2] دو رشته بیت M و N را با $M||N$ نمایش می‌دهیم.

مجموعه رشته‌بیت‌ها شامل رشته تهی و بدون رشته تهی را به‌ترتیب با Z_2^* و Z_2^+ نمایش می‌دهیم. مجموعه رشته‌بیت‌ها با طول نامحدود را با Z_2^∞ نمایش می‌دهیم.

۲-۲- قوانین دنباله‌زنی

برای قوانین دنباله‌زنی[3] از نمادگذاری زیر استفاده می‌کنیم: دنباله‌زنی پیام M به یک دنباله از قالب‌های x-بیتی را با $M||pad[x](|M|)$ نمایش می‌دهیم. برای دنباله‌زنی‌های یک‌به‌یک، عنوان دنباله زدایی[4] را برای بازیابی M از $P = M||pad[x](|M|)$ استفاده می‌کنیم.

یک دنباله‌زنی، منطبق بر اسفنج است اگر هرگز رشته تهی نتیجه ندهد و در شرط زیر صدق کند:

$$\forall n \geq 0, \forall M, M' \in Z_2^*: M \neq M' \Rightarrow M \| pad[r](|M|) \neq M' \| pad[r](|M'|) \| 0^{nr}$$

۲-۳- ساختار اسفنجی

ساختار اسفنجی، تابع $SPONGE[f, pad, r]$ را با دامنه Z_2^* و برد Z_2^∞ می‌سازد که در آن f تبدیل یا جایگشت

[1] Output binding

[2] Concatenation
[3] Padding
[4] Unpadding

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

bبیتی، pad دنباله‌زنی منطبق بر اسفنج و r پارامتر نرخ بیتی است (شکل ۱).

ساختار اسفنجی یک حالت داخلی bبیتی به‌نام s دارد. ابتدا تمام بیت‌های حالت داخلی با صفر مقداردهی و پیام ورودی دنباله‌زنی شده و به قالب‌های rبیتی بریده می‌شود؛ سپس دو فاز اجرا می‌شود: فاز جذب‌کردن[1] و فاز فشرده‌شدن[2]. در این فازها، با r بیت ابتدایی و باقی b-r بیت از حالت داخلی متفاوت رفتار می‌شود. r بیت ابتدایی حالت داخلی را بخش بیرونی[3] نامیده و آن را با $\bar{s}$ نمایش می‌دهیم. همچنین، b-r بیت باقی از حالت داخلی s را بخش درونی[4] نامیده و آن را با $\hat{s}$ نمایش می‌دهیم. بخش درونی حالت داخلی s، ظرفیت c نامیده می‌شود. دو فاز یادشده به‌شرح زیر است:

فاز جذب‌کردن: قالب‌های r-بیتی ورودی با بخش بیرونی حالت داخلی XOR شده و پس از جذب هر قالب rبیتی تابع f اجرا می شود. زمانی‌که تمام قالب‌های پیام پردازش(جذب) شد، ساختار اسفنجی وارد فاز فشرده‌شدن می‌شود.

فاز فشرده‌شدن: بخش بیرونی حالت داخلی به‌عنوان خروجی برگردانده شده و پس از تولید هر r بیت خروجی، تابع f روی حالت داخلی اسفنج اجرا می‌شود. تعداد دفعات تکرار با توجه به طول خروجی درخواستی l تعیین و درنهایت خروجی اسفنج به‌عنوان خروجی معرفی می‌شود. c بیت حالت داخلی هرگز به‌طور مستقیم از قالب‌های ورودی تأثیر نگرفته و هرگز در فاز فشرده‌شدن به‌طور مستقیم به‌عنوان خروجی معرفی نمی‌شود. ظرفیت c تعیین‌کنندهٔ میزان امنیت قابل حصول ساختار است.

به‌روزرسانی حالت داخلی با استفاده از تابع f که یک تبدیل تصادفی و یا یک جایگشت تصادفی است، انجام می‌شود. یک تبدیل تصادفی با طول داده‌شده b، تبدیلی است که به‌صورت تصادفی و یک‌نواخت از میان 2^{b2^b} تبدیل b-بیتی انتخاب شده است. یک جایگشت تصادفی با طول داده‌شده b، جایگشتی است که به‌صورت تصادفی و یک‌نواخت از میان $2^b!$ جایگشت bبیتی انتخاب شده است.

۲-۴- توابع کمکی

تابع کمکی اول، تابع جذب $ABSORB[f,r]$ است. این تابع رشته P با $|P|$ ضریبی از r را به‌عنوان ورودی گرفته و حالت داخلی را بعد از جذب P برمی‌گرداند (شکل ۲). برای دنباله P، $s = absorb(P)$ معرف حالت داخلی اسفنج پس از جذب دنباله P است. P را مسیری به وضعیت s می‌گوییم، اگر $s = absorb(P)$ باشد.

تابع جذب کردن $ABSORB[f,r]$

۱. شرط: r<b

۲. واسط: $P \epsilon \mathbb{Z}_{2r}^{*}$ و $s \epsilon \mathbb{Z}_2^b$ که $s = absorb(P)$

۳. حالت داخلی s را مقداردهی کن $s = 0^b$

۴. به ازای i از صفر تا $|P|_r - 1$ مراحل ۵ و ۶ را تکرار کن

۵. $s = s \oplus (P_i || 0^{b-r})$

۶. $s = f(s)$

۷. s را بعنوان خروجی برگردان

(شکل-۲): الگوریتم تابع جذب‌کردن

تابع کمکی که از برخی جهات دوگان تابع جذب‌کردن است، تابع فشردن $SQUEEZE[f,r]$ است. برای یک حالت داخلی داده‌شده s، $squeeze(s,l)$ معرف خروجی بریده به طول l بیت از تابع اسفنجی، با حالت داخلی s در ابتدای فاز فشردن، است (شکل ۳).

تابع فشردن $SQUEEZE[f,r]$

۱. شرط: r<b

۲. واسط: $Z \epsilon \mathbb{Z}_2^l$ و $s \epsilon \mathbb{Z}_2^b$ که $Z = squeeze(s,l)$ و $l > 0$

۳. Z را مقداردهی کن $Z = |s|_r$

۴. تا زمانیکه $|Z|_r r < l$ مراحل ۵ و ۶ را تکرار کن

۵. $s = f(s)$

۶. $Z = Z || |s|_r$

۷. $|Z|_l$ را بعنوان خروجی برگردان

(شکل-۳): الگوریتم تابع فشردن

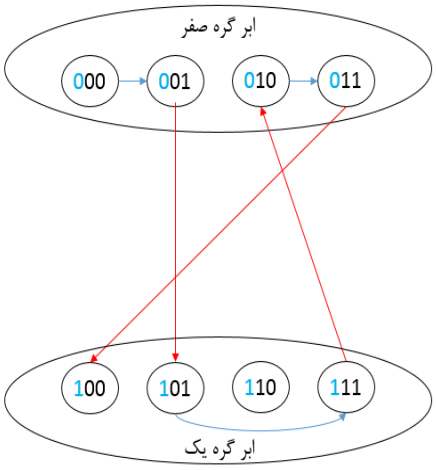
۲-۵- نمایش گرافیکی یک تابع اسفنجی

یک تابع اسفنجی را با گرافی با $2^b = 2^{r+c}$ گره و 2^b شاخه متناظر کرده و آن را گراف اسفنج می‌نامیم. گره‌ها مقادیر حالت داخلی اسفنج بوده و برای هر زوج (s,t) که $t = f(s)$، یک شاخه مستقیم از s به t وجود دارد. از هر گره به‌طوردقیق یک شاخه خارج می‌شود. اگر f یک جایگشت باشد، به هر گره یک شاخه وارد می‌شود؛ در غیر این‌صورت، ممکن است به هر گره بیش از یک شاخه وارد شود. گره‌ها بر اساس بخش درونی حالت داخلی قابل افراز بوده و زیرمجموعه گره‌های با مقدار یکسان برای بخش درونی حالت داخلی را یک ابرگره می‌نامیم؛ بنابراین شاخه‌های بین گره‌ها شاخه‌های بین ابرگره‌ها نیز هستند. درکل 2^c ابرگره، یک ابرگره برای هر مقدار ممکن برای

[1] Absorbing phase
[2] Squeezing phase
[3] Outer part
[4] Inner part

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

بخش درونی حالت داخلی، وجود دارد. 2^r گره داخل یک ابرگره با بخش بیرونی حالت داخلی اسفنج $\bar{s}$ مشخص و تعیین می‌شود. مثالی از نمایش گرافیکی تابع اسفنجی با r=2 و c=1 و جایگشت f(x) به‌صورت زیر در شکل (۴) نشان داده شده است:



(شکل-۴): نمایش گرافیکی تابع اسفنجی با r=2 و c=1 و رشته ورودی P=011001

x	000	001	010	011	100	101	110	111
$f(x)$	011	101	111	100	001	110	000	010

رشته ورودی P با دنبال‌کردن شاخه‌ها از ابرگره 0 (ریشه) جذب می‌شود. ابتدا یک شاخه از $P_0||0^c$ رسم می‌کنیم؛ این شاخه به گره‌ای با مقدار بخش بیرونی حالت داخلی $\overline{absorb}(P_0)$ از ابرگره $\widehat{absorb}(P_0)$ می‌رسد، آن‌گاه شاخه‌ای از گره‌ای داخل آن ابرگره با بخش بیرونی حالت داخلی $\overline{absorb}(P_0) \oplus P_1$ به گرهٔ $absorb(P_0||P_1)$ می‌رسد. برای P_i، شاخه‌ای از گره با مقدار بخش بیرونی حالت داخلی $\overline{absorb}(P_0||P_1 \dots ||P_{i-1}) \oplus P_i$ داخل ابرگره $\widehat{absorb}(P_0||P_1 \dots ||P_{i-1})$ ترسیم می‌کنیم؛ درنتیجه نمایش گرافیکی یک مسیر به یک حالت داخلی دنباله‌ای از شاخه‌های جهت‌دار از ریشه به ابرگره متناظر است. درصورتی‌که نمایش گرافیکی معلوم باشد، مقدار P قابل بازیابی است. قالب i-ام از مسیر P با شاخه‌های منتهی به و شروع‌شونده از ابرگره i-ام در مسیر قابل تعیین است. این قالب حاصل XOR مقدار بخش بیرونی حالت داخلی گره‌ای که شاخه بیرون رونده آغاز می‌شود با مقدار بخش بیرونی حالت داخلی گره‌ای که شاخه وارد شونده به آن می‌رسد است. نخستین قالب از مسیر P_0 متناظر با ریشه که هیچ شاخه‌ای وارد نمی‌شود و برابر با بخش بیرونی حالت داخلی گره‌ای که نخستین شاخه آغاز می‌شود، است.

۲-۶- مدل مهاجم

مدل درنظرگرفته شده برای مهاجم به‌صورت زیر است. در ابتدا، مهاجم اطلاعاتی در مورد f ندارد. تنها راهی که وی می‌تواند اطلاعاتی در مورد f به‌دست آورد، فراخوانی f (و f^{-1} درصورتی‌که f یک جایگشت باشد) است.

این مدل متناظر با دنیای واقعی است که در آن مهاجم مشخصات f را داشته و بهینه‌ترین راه برای محاسبه $f(x)$ (یا $f^{-1}(x)$) برای یک x مشخص، اجرای برنامه‌ای است که این مقادیر را محاسبه می‌کند. در این مدل فراخوانی f در پیش محاسبات نیز در محاسبه پیچیدگی در نظر گرفته می‌شود.

در ادامه، اطلاعاتی را که دشمن در آزمایش به‌دست آورده است، در گرافی که متناظر با بخش‌های معلوم گراف اسفنج برای مهاجم است، نمایش می‌دهیم. این گراف را گراف مهاجم می‌نامیم.

در ابتدا، گراف مهاجم هیچ شاخه‌ای ندارد. بدون تغییر در جامعیت مسأله، فرض می‌کنیم مهاجم پرسمانی برای شاخه‌های مشخص گراف انجام نمی‌دهد؛ بنابراین، فراخوانی f متناظر با افزودن شاخه‌ای شروع‌شونده از یک گره مشخص و فراخوانی f^{-1} متناظر با افزودن یک شاخه وارد شونده به یک گره مشخص در گراف مهاجم است.

در گراف مهاجم، اگر دنباله‌ای از شاخه‌های جهت‌دار از ابرگره $\hat{s}$ به ابرگره $\hat{t}$ (در جهت صحیح) وجود داشته باشد و یا $t = s$، ابرگره $\hat{t}$ از ابرگره $\hat{s}$ دسترس‌پذیر خواهد بود. ابرگره‌های دسترس پذیر از ریشه را ابرگره ریشه‌دار[1] نامیده و مجموعه آنها را با $\mathcal{R}$ نمایش می‌دهیم ($R = |\mathcal{R}|$). همچنین تمامی گره‌های داخل یک ابرگره ریشه‌دار را ریشه‌دار می‌نامیم.

۲-۷- تابع هزینه

برای احتمال موفقیت بهینه انجام حمله عبارت $Pr(success)$ را به‌صورت تابعی از N محاسبه می‌کنیم، که N تعداد دفعات فراخوانی f توسط مهاجم زمانی‌که f یک تبدیل است و مجموع تعداد دفعات فراخوانی f و f^{-1} زمانی‌که f یک جایگشت است. این احتمال برابر با تعداد تبدیل‌ها (یا جایگشت‌ها)ی f که حمله موفق بوده تقسیم بر مجموع تعداد تبدیل‌ها (یا جایگشت‌ها)ی f با ابعاد

[1] Rooted supernode

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

مشخص است. بنابراین، احتمال موفقیت ۱٪ بدان معناست که برای ۹۹٪ فراخوانی‌های f، حمله موفق نیست.

عبارت $Pr(success)$ برای حملات اولیه مختلف به صورت $1-e^{\nu(N)}$ است، که $\nu(N)$ چندجمله‌ای بر حسب N از درجه یک یا دو می‌باشد. برای ساده‌سازی نگارش، تابع هزینه $c_P(N)$ یک حمله را به‌صورت $c_P(success)=-log(1-Pr(success))$ تعریف می‌کنیم؛ بدین ترتیب داریم:

$$Pr(success)=1-e^{-c_P(success)}$$

برای مقادیر N که $c_P(success)\ll 1$ می‌توانیم از تقریب $log(1+\epsilon)$ استفاده کرده و خواهیم داشت:

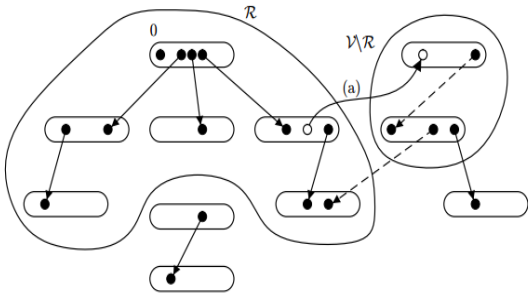
$$Pr(success)\approx c_P(success)$$

در ادامه مقاله به بررسی ویژگی‌های امنیتی ساختارهای مبتنی بر اسفنج می‌پردازیم. با استفاده از تحلیل‌های پیش رو، طراح امکان می‌یابد تا پارامترهای الگوریتم خود را انتخاب و کران‌های امنیتی مورد نظر را پیشنهاد دهد. برای نمونه، چنانچه حمله‌کننده توانایی جهت یافتن برخورد در حالت داخلی الگوریتم داشته باشد، می‌تواند به فراخور آن‌که الگوریتم پیشنهادی یک تابع درهم‌ساز یا یک رمز دنباله‌ای است، حملاتی را بر الگوریتم اعمال کند؛ بنابراین بخش‌های پیش رو می‌تواند به‌کمک طراحان و تحلیل‌گران الگوریتم‌های مبتنی بر ساختار اسفنج آید.

۸-۲- ایجاد برخورد داخلی

در این بخش به معرفی نحوه مؤثر یافتن برخورد داخلی در ساختار اسفنجی و پیچیدگی آن می‌پردازیم. این روش تحلیل معادل حمله برخورد در توابع چکیده‌ساز است و زمانی که از ساختار اسفنجی برای طراحی یک تابع چکیده‌ساز استفاده شود از اهمیت فراوانی برخوردار خواهد بود.

اگر مهاجم دو مسیر از ریشه به یک ابرگره بیابد یک برخورد داخلی خواهد داشت. ما i-امین فراخوانی توسط مهاجم را در نظر گرفته و احتمال رسیدن به برخورد داخلی را در شرایطی که هیچ برخوردی تاکنون یافت نشده است، بیان می‌کنیم. همان‌طور که در شکل (۵) دیده می‌شود، این بدان معناست که بایستی شاخه جدید یک ابرگره ریشه‌دار را به ابرگره‌ای که از آن یک ابرگره ریشه‌دار دسترس‌پذیر باشد، متصل کند. در گراف مهاجم، ابرگره‌ای را که از آن یک ابرگره ریشه‌دار دسترس‌پذیر باشد، یک ابرگره $\mathcal{R}$-دسترس‌پذیر و مجموعه آنها را $\mathcal{V}$ می‌نامیم ($V=|\mathcal{V}|$). واضح است که $\mathcal{R}\subseteq\mathcal{V}$ در ابتدا، $\mathcal{V}=\mathcal{R}=\{0\}$ و $R=V=1$. درست پیش از افزودن شاخه i-ام، گراف شامل i-1 شاخه بوده و $R\le V\le i$.



(شکل-۵): افزودن شاخه (a) یک برخورد درونی را نتیجه می‌دهد. شاخه (a) بایستی از R شروع شده و به v ختم شود [۱].

۹-۲- ایجاد برخورد داخلی با فرض f یک تبدیل تصادفی

مهاجم تنها می‌تواند شاخه‌هایی را که از گره‌های مشخص شروع می‌شود، اضافه کند. اگر شاخه جدید از یک گره ریشه‌دار شروع شود، احتمال موفقیت $V/2^c$ است، علاوه‌براین، یک واحد به R و به تبع آن به V افزوده خواهد شد. اگر شاخه جدید از یک گره بدون ریشه[1] شروع شود، احتمال موفقیت صفر است. مقدار R بدون تغییر مانده و درصورتی‌که به گره‌ای در $\mathcal{V}$ برسد، ممکن است مقدار V را یک واحد افزایش دهد. این امر نتیجه می‌دهد احتمال موفقیت شاخه‌های بعدی همیشه با افزودن شاخه‌هایی که از گره‌های ریشه‌دار شروع می‌شود، بهینه می‌شود. شکل دقیق درخت ریشه‌دار اهمیت ندارد؛ بنابراین، با به‌کارگیری این روش، درست پیش از افزودن شاخه i-ام، R=V=i و داریم:

$$Pr(no\ IC)=\prod_{i=1}^{N}(1-\frac{i}{2^c})$$

اگر $N\ll 2^c$ باشد، می‌توان از تقریب $log(1+\epsilon)$ استفاده کرده و داریم:

$$Pr(IC)\approx 1-e^{-\sum_{i=1}^{N}\frac{i}{2^c}}=1-e^{-\frac{N(N+1)}{2^{c+1}}}$$

بنابراین تابع هزینه به‌صورت زیر است:

$$c_P(IC)\approx\frac{N(N+1)}{2^{c+1}}$$

[1] Non-rooted

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]
دوفصل‌نامه علمی ترویجی اطلاعات تبادل تولید و امنیت فضای

۱۰-۲- ایجاد برخورد داخلی با فرض f یک جایگشت تصادفی

اگر f یک جایگشت باشد، مهاجم می‌تواند شاخه‌هایی را که از گره‌های منتخب آغاز می‌شود و شاخه‌هایی که به گره‌های منتخب ختم می‌شود، اضافه کند، علاوه بر این، یک شاخه شروع‌شده از یک گره منتخب می‌تواند تنها به گره‌ای برسد که شاخه‌ای تا کنون به آن وارد نشده باشد. شاخهٔ واردشونده به یک گره منتخب تنها می‌تواند از گره‌ای آغاز شود که شاخه‌ای از آن خارج نشده باشد.

اگر شاخه‌ای که از یک گره منتخب ریشه‌دار آغاز می‌شود، اضافه گردد، احتمال موفقیت برابر با تعداد گره‌های بدون شاخه واردشونده در ν تقسیم بر کل تعداد گره‌های بدون شاخه واردشونده است:

$$\frac{(2^r-1)V+1}{2^{r+c}-i}$$

اگر یک شاخه واردشونده به یک گره منتخب در ν اضافه شود، احتمال موفقیت به‌صورت مشابه برابر است با:

$$\frac{(2^r-1)R+1}{2^{r+c}-i}$$

هرچه مقادیر R و V بزرگ‌تر باشد، احتمال موفقیت در پرسمان‌های بعدی بالاتر خواهد بود؛ بنابراین می‌توان با انجام پرسمان از انواع دیگر مقادیر R و V را سریع‌تر افزایش داد. شاخه آغازشونده از یک گره منتخب که ریشه‌دار نیست، نمی‌تواند به برخورد منجر شود. این شاخه مقدار R را تغییر نداده و ممکن است مقدار V را یک واحد افزایش دهد. اما یک شاخه وارد شونده به یک گره منتخب در ν یک واحد به V اضافه کرده و همواره احتمال موفقیت بهتری را نتیجه می‌دهد. بطور مشابه، شاخه واردشونده به یک گره منتخب که در ν نیست منجر به برخورد نمی‌شود؛ اما می‌تواند یک واحد به مقدار R اضافه کند. شاخه آغازشونده از یک گره ریشه‌دار منتخب به‌طور حتم یک واحد به R اضافه کرده و همواره احتمال موفقیت بهتری را نتیجه می‌دهد.

همواره رابطه $R \le V \le i$ برقرار است. به‌طور کلی، استراتژی بهینه آن است که در آن احتمال موفقیت i-امین فراخوانی برابر است با:

$$\frac{(2^r-1)i+1}{2^{r+c}-i}$$

با افزودن شاخه واردشونده به یک گره منتخب در ν که به برخورد منجر نمی‌شود، R تغییر نکرده و R<i نتیجه می‌شود، درحالی‌که در استراتژی بهینه R=i است؛ بنابراین، در استراتژی بهینه تنها یک شاخه واردشونده به یک گره منتخب در ν اضافه شده و تمامی سایر شاخه‌ها تنها شاخه‌های اضافه‌شده به گره‌های ریشه‌دار هستند. بدین ترتیب داریم:

$$Pr(no\ IC) = \prod_{i=1}^{N}\left(1-\frac{(2^r-1)i+1}{2^{r+c}-i}\right) = \prod_{i=1}^{N}\frac{1-\frac{i}{2^c}-\frac{1}{2^{r+c}}}{1-\frac{i}{2^{r+c}}}$$

با استفاده از تقریب $log(1+\epsilon)$ داریم:

$$c_P(IC) \approx \sum_{i=1}^{N} -\frac{i-1}{2^{r+c}}+\frac{i}{2^c} = \frac{N(N+1)}{2^{c+1}}-\frac{N(N-1)}{2^{r+c+1}}$$

۳- یافتن مسیر به یک حالت داخلی

روش مؤثر برای یافتن مسیر به یک حالت داخلی در ساختار اسفنجی و پیچیدگی آن در این بخش تشریح می‌شود. این روش تحلیل را می‌توان معادل حمله پیش‌تصویر در توابع چکیده‌ساز در نظر گرفت. در این تحلیل، برای بخش درونی مشخص $\hat{t}$ از حالت داخلی t، مهاجم بایستی مسیر p را بیابد به‌گونه‌ای که $\widehat{absorb}(p) = \hat{t}$. i-امین فراخوانی مهاجم را در نظر گرفته و احتمال رسیدن به مسیر مورد نظر را در شرایطی که هیچ مسیری تا کنون یافت نشده است، مشخص می‌کنیم. همان‌طور که در شکل (۶) دیده می‌شود، این بدان معناست که شاخه جدید بایستی یک ابرگره ریشه‌دار را به ابرگره‌ای که $\hat{t}$ از آن دسترس‌پذیر باشد، متصل کند. ابرگره (و گره‌هایی) را که از آن هدف مورد نظر قابل حصول است، ابرگره (گره‌های) به هدف نائل‌شونده[1] نامیده و مجموعه آنها را با ν نمایش می‌دهیم، $V = |\nu|$. در ابتدا، $\nu = \{\hat{t}\}$، $\mathcal{R} = \{0\}$ و R=V=1. به‌طوردقیق پیش از i-امین فراخوانی، گراف شامل i-1 شاخه بوده و $R \le i$، $V \le i$ و $R + V \le i + 1$.

۱-۳- یافتن مسیر به یک حالت داخلی با فرض f یک تبدیل تصادفی

مهاجم تنها می‌تواند شاخه‌هایی را اضافه کند که از گره‌های منتخب آغاز می‌شوند. اگر شاخه شروع شونده از گره منتخب ریشه‌دار اضافه شود، احتمال موفقیت برابر با $V/2^c$ است. در غیر این‌صورت، احتمال موفقیت برابر صفر بوده، R بدون تغییر مانده و یک واحد به V با احتمال

[1] Target-reaching

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

$V/2^c$ اضافه می‌شود. این مسأله نتیجه می‌دهد که بهترین روش برای بهینه‌کردن احتمال موفقیت افزودن شاخه‌های آغازشونده از گره‌های منتخب ریشه‌دار است؛ بنابراین، با به‌کاربردن این روش، به‌طوردقیق پیش از فراخوانی i-ام، R=i و V=1 بوده و داریم:

$$Pr(no\ path) = \prod_{i=1}^{N}(1 - \frac{1}{2^c})$$

با استفاده از تقریب $log(1+\epsilon)$ برای $1 \gg 2^c$ داریم:

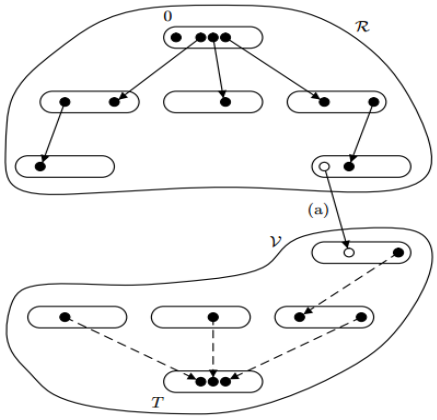
$$c_p(path) \approx \frac{N}{2^c}$$

حال نسخهٔ دیگری از حمله را در نظر می‌گیریم: پیداکردن یک مسیر ثانوی به یک حالت داخلی درصورتی‌که در حال حاضر یک مسیر به طول l وجود داشته باشد. این نسخه از حمله مربوط به پیش‌تصویر دوم در زمان استفاده از ساختار بعنوان یک تابع چکیده ساز است. احتمال یافتن یک مسیر ثانوی پس از افزودن N شاخه (با درنظرگرفتن l شاخه متناظر با جذب پیام p) را شمارش می‌کنیم. پس از افزودن این l شاخه، $\mathcal{R}$ و $\mathcal{V}$ هر یک شامل مجموعه l ابرگره در مسیر از ریشه به t هستند. برای N>l نتیجه می‌شود:

$$Pr(no\ 2nd\ path) = \prod_{i=l}^{N}(1 - \frac{l}{2^c})$$

و متعاقبا، اگر $l \ll 2^c$

$$c_p(2nd\ path) \approx \frac{l(N-l)}{2^c}$$



(شکل-۶): افزودن شاخه (a) یک مسیر به یک حالت داخلی را نتیجه می‌دهد. شاخه (a) بایستی از $\mathcal{R}$ شروع شده و به $\mathcal{V}$ ختم شود [۱].

۲-۳- یافتن مسیر به یک حالت داخلی با فرض f یک جایگشت تصادفی

در این نوع تحلیل، مهاجم می‌تواند شاخه‌های شروع‌شونده از گره‌های انتخابی و نیز شاخه‌های واردشونده به گره‌های انتخابی را به گرافش اضافه کند. یک شاخه شروع‌شده از یک گره منتخب می‌تواند تنها به گره‌ای برسد که شاخه‌ای تا کنون به آن وارد نشده باشد. شاخه‌ی وارد شونده به یک گره منتخب تنها می‌تواند از گره‌ای آغاز شود که شاخه‌ای از آن خارج نشده باشد.

اگر شاخه از یک گره منتخب ریشه‌دار آغاز شده و ابرگره‌های $\mathcal{V}$ به همراه شاخه‌ها تشکیل یک درخت بدهند، احتمال موفقیت برابر است با تعداد گره‌های بدون شاخه واردشونده در $\mathcal{V}$ تقسیم بر تعداد کل گره‌های بدون شاخه واردشونده، یعنی:

$$\frac{(2^r-1)V+1}{2^{r+c}-i}$$

اگر برخورد داخلی وجود نداشته باشد، این عمل یک واحد به R اضافه کرده و V بدون تغییر می‌ماند.

اگر یک شاخه واردشونده به گره به هدف نائل‌شونده منتخب اضافه شود، به‌طور مشابه احتمال موفقیت برابر است با:

$$\frac{(2^r-1)R+1}{2^{r+c}-i}$$

با این فرض که برخورد داخلی وجود ندارد. این عمل، اگر شاخه جدید از یک گره به هدف نائل‌شونده شروع نشود، یک واحد به V اضافه کرده و R بدون تغییر می‌ماند. فرض می‌کنیم برخورد درونی وجود نداشته و ابرگره‌های $\mathcal{V}$ تشکیل یک درخت داده و بعدتر صحت این فرض را بررسی می‌کنیم.

هرچه مقادیر R و V بیشتر باشد، احتمال موفقیت در پرسمان‌های بعدی بیشتر خواهد بود. این امر نتیجه می‌دهد افزودن یک شاخه واردشونده به گره‌های به هدف نائل شونده، اگر در ادامه شاخه شروع شونده از گره‌های ریشه‌دار افزوده شود (و بالعکس)، احتمال موفقیت را افزایش می‌دهد.

شاخه شروع شونده از یک گره منتخب که ریشه‌دار نیست به مسیری به $\hat{t}$ منجر نخواهد شد. این عمل مقدار R را تغییر نداده و با احتمال کمی یک واحد به V اضافه می‌کند. با نگاهی به افزایش احتمال موفقیت در پرسمان‌های بعدی، افزودن یک شاخه شروع‌شونده از یک گره ریشه‌دار همیشه بهتر است. به‌طور مشابه، شاخه واردشونده به یک گره منتخب که در $\mathcal{V}$ نیست به مسیری به $\hat{t}$ منجر نخواهد شد. این عمل، با احتمال کمی یک واحد به R اضافه کرده و افزودن یک شاخه واردشونده به یک گره به هدف نائل‌شونده همواره بهتر است.

ما متغیر δ_i را تعریف می‌کنیم؛ بدین صورت که اگر شاخه i-ام اضافه‌شده از یک گره منتخب ریشه‌دار آغاز شود مقدار آن برابر ۱ و در غیر اینصورت مقدار آن برابر

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

۱- خواهد بود. مقدار R و V را به‌طوردقیق قبل از اضافه کردن شاخه i-ام با R_i و V_i نشان می‌دهیم. آن‌گاه احتمال این‌که اضافه‌کردن شاخه i-ام مسیری را نتیجه ندهد، برابر است با:

$$1-\frac{(2^r-1)\left(\frac{1+\delta_i}{2}V_i+\frac{1-\delta_i}{2}R_i\right)+1}{2^{r+c}-i}$$
$$=\frac{1-\frac{i+1}{2^{r+c}}-\frac{2^r-1}{2^{r+c+1}}(V_i+R_i-\delta_i(R_i-V_i))}{1-\frac{i}{2^{r+c}}}$$

با استفاده از تقریب $log(1+\varepsilon)$ داریم:

$$c_p(path)\approx\sum_{i=1}^{N}(\frac{1}{2^{r+c}}+\frac{2^r-1}{2^{r+c+1}}(V_i+R_i-\delta_i(R_i-V_i)))$$

رابطه $V_i+R_i\le i+1$ برقرار بوده و تساوی مربوط به زمانی است که برخورد درونی وجود نداشته و ابرگره‌های $\mathcal{V}$ تشکیل یک درخت بدهند، ابتدا فرض می‌کنیم $V_i+R_i=i+1$ و در ادامه صحت این فرض را بررسی می‌کنیم؛ علاوه‌براین داریم:

$$R_i-V_i=\sum_{j=1}^{i-1}\delta_j$$

بدین ترتیب

$$c_p(path)\approx\frac{N}{2^{r+c}}+\frac{2^r-1}{2^{r+c+2}}(N^2+3N+1-\sum_{i=1}^{N}\sum_{j=1}^{i-1}2\delta_i\delta_j)$$

برای جمله آخر داریم:

$$\sum_{i=1}^{N}\sum_{j=1}^{i-1}2\delta_i\delta_j=\sum_{i=1}^{N}\sum_{j=1}^{N}\delta_i\delta_j-\sum_{i=1}^{N}\delta_i\delta_i=(R_{N+1}-V_{N+1})^2-N$$

بنابراین داریم:

$$c_p(path)\approx\frac{N(N+4)-(R_{N+1}-V_{N+1})^2}{2^{c+2}}-\frac{N^2-(R_{N+1}-V_{N+1})^2}{2^{r+c+2}}$$

اگر برای Nهای زوج $R_{N+1}=V_{N+1}$ و برای Nهای فرد $|R_{N+1}-V_{N+1}|=1$، بدین معنا که $\mathcal{R}$ و $\mathcal{V}$ به‌طوردقیق قبل از پیداشدن مسیر تعداد گره‌های یکسانی داشته باشند، مقدار $c_p(path)$ بیشینه خواهد شد. از آنجا که مشخص‌شدن مسیر مورد نظر پیشاپیش معلوم نیست، بهترین روش افزودن شاخه‌های شروع‌شونده از گره‌های منتخب در $\mathcal{R}$ و شاخه‌های واردشونده به گره‌های منتخب در $\mathcal{V}$ در مد یک در میان با شرط $(R_N-V_N)^2\le 1$ است. در این‌صورت فرض $V_i+R_i=i+1$ صحیح بوده و برای Nهای فرد داریم:

$$c_p(path)\approx\frac{N(N+4)}{2^{r+c}}-\frac{N^2}{2^{r+c+2}}$$

زمانی‌که N از مرتبه $2\sqrt{2^c}$ و در نتیجه R و V از مرتبه $\sqrt{2^c}$ است؛ احتمال موفقیت قابل توجه خواهد شد. این بدان معناست که برای این مقادیر از N ممکن است برخورد درونی وجود داشته باشد؛ اما تعداد کم آنها در مقایسه با R تأثیر قابل توجهی در احتمال موفقیت نخواهد داشت.

۴- تشخیص دور[1] در خروجی

در این بخش به معرفی نحوه مؤثر یافتن دور در خروجی ساختار اسفنجی و پیچیدگی آن می‌پردازیم. این روش تحلیل زمانی که از ساختار اسفنجی برای طراحی یک رمز جریانی و یا یک مولد اعداد شبه‌تصادفی استفاده شود از اهمیت فراوانی برخوردار خواهد بود.

هدف این روش تحلیل یافتن دورهایی در خروجی، متناظر با رشته ورودی‌های معتبر، است. مهاجم می‌تواند یک رشته ورودی P را گرفته و با اعمال آن به اسفنج با عمل جذب‌کردن، گره $absorb(P)$ را تعیین کند. از این گره، قالب‌های خروجی $\overline{absorb}(P||0^{jr})$ با دنبال‌کردن زنجیره گره‌های متصل با شاخه‌ها ساخته می‌شود، بدین ترتیب که $absorb(P||0^{jr})=f(absorb(P||0^{(j-1)r}))$؛ زنجیره را به‌صورت دنباله‌ای از گره‌های متصل با شاخه‌های جهت‌دار تعریف می‌کنیم. نخستین گره در زنجیره گره $u=absorb(P')\oplus(P_{|P|_r-1}||0^c)$ است که در آن P' همان P است که قالب آخر $(P_{|P|_r-1})$ حذف شده است.

مهاجم با توسیع در انتها با افزودن شاخه تا رسیدن به گره‌ای در زنجیره به یک دور دست خواهد یافت. کوتاه‌ترین مسیر رشته ورودی معتبر شامل یک قالب ناصفر است. پیش از افزودن شاخه i-ام، زنجیره شامل i گره است.

۴-۱- تشخیص دور در خروجی با فرض f یک تبدیل تصادفی

احتمال آن‌که شاخه جدید به یکی از گره‌های زنجیره وارد شود، برابر با $i/2^{r+c}$ است. با استفاده از تقریب $log(1+\varepsilon)$ داریم:

$$c_P(output\ cycle)\approx\frac{N(N+1)}{2^{r+c+1}}$$

[1] Cycle

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

۴-۲- تشخیص دور در خروجی با فرض f یک جایگشت تصادفی

هر زمان، تنها یک گره در زنجیره وجود دارد که هیچ شاخه‌ای به آن وارد نمی‌شود، گره u. بنابراین، احتمال این‌که شاخه جدید به گره‌ای در زنجیره وارد شود، برابر با $1/2^{r+c}$ است. بنابراین داریم:

$$c_P(output\ cycle) \approx \frac{N}{2^{r+c}}$$

۴-۳- بازیابی حالت داخلی

بازیابی حالت داخلی[1] شامل یافتن حالت داخلی s برای رشته داده‌شده $Z = squeeze(s, |Z|)$ است. یکی از کاربردهای این روش تحلیل ارزیابی امنیت رمزهای جریانی مبتنی بر ساختار اسفنجی و تعیین پیچیدگی یافتن حالت داخلی بازای یک رشته‌داده خروجی (دنباله کلید اجرایی[2]) مشخص است.

۴-۴- بازیابی حالت داخلی با فرض f یک تبدیل تصادفی

مهاجم می‌تواند حدس‌های a را برای $\hat{s}$ در نظر گرفته و صحت آن‌ها را با پرسمان از f بررسی کند. احتمال موفقیت بعد از n حدس برابر با $n2^{-c}$ است. بررسی صحت یک حدس به روش زیر قابل انجام است. یک پرسمان به‌صورت $a_1 = f(a_0||Z_0)$ ارسال شده و تطابق بخش بیرونی نتیجه a_1 با Z_1 بررسی می‌شود. در صورت برقراری تطابق، مهاجم پرسمان $a_2 = f(a_1)$ را انجام داده و تساوی بخش بیرونی نتیجه a_2 را با Z_2 بررسی کند، و به همین ترتیب تا آخر. تعداد پرسمان‌های مورد انتظار برای یک حدس غلط برابر است با:

$$1 + 2^{-r} + 2^{-2r} + \cdots \approx \frac{1}{1 - 2^{-r}}$$

بنابراین احتمال موفقیت مورد انتظار بعد از N پرسمان برابر است با :

$$N\frac{1 - 2^{-r}}{2^c}$$

۴-۵- بازیابی حالت داخلی با فرض f یک جایگشت تصادفی

در این قسمت فرض می‌کنیم $|Z|$ ضریبی از نرخ بیتی است. پاسخ s را با s_0، $f(s_0)$ را با s_1 و $f(s_i)$ را با s_{i+1} نمایش می‌دهیم.

[1] State recovery
[2] Keystream

وقتی f یک جایگشت است، مهاجم می‌تواند ابتدا $\hat{s_i}$ را به‌ازای مقداری برای اندیس i حدس زده و سپس مقدار $\hat{s_0}$ را از s_i با تکرار به‌کارگیری f^{-1} محاسبه کند. این مسأله روی احتمال موفقیت تأثیرگذار است.

مهاجم غیرفعال

ابتدا تقسیم قالب مستقیم و معکوس رشته Z با $|Z| = mr$ و تعدد مستقیم و معکوس را تعریف می‌کنیم. تقسیم قالب مستقیم[3] $B_f(Z)$ قسمتی از اندیس‌های قالب i از Z با $0 \le i < |Z|_r - 1$ است که بر اساس مقادیر Z_i گروه-بندی شده‌اند. زیرگروه‌های $B_f(Z)$ را با $B_{(j)}$ و Z_i متناظرشان را با $Z_{(j)}$ نمایش می‌دهیم؛ بنابراین داریم:

$$\forall i \in B_{(j)}: Z_i = Z_{(j)}$$

و

$$\forall i \notin B_{(j)}: Z_i \neq Z_{(j)}.$$

توجه کنید که در تقسیم قالب مستقیم، اندیس مربوط به آخرین قالب در نظر گرفته نمی‌شود. تعدد مستقیم[4] رشته Z، با $m_f(Z,r)$ نشان داده شده و برابر با اندازه بزرگ‌ترین زیرگروه $B_f(Z)$ است. به بیان دیگر، تعدد مستقیم بیان‌گر تعداد دفعات تکرار قالب Z_i است که بیشترین تکرار را در Z دارد.

اگر r بزرگ باشد، برای یک دنباله تصادفی با $|Z|_r < 2^{r/2}$ تعدد مستقیم به‌طورعمومی برابر با یک است؛ بدین معنی که تمام قالب‌های Z_i از Z متفاوت هستند. اگر r=1، قالب‌ها بیت هستند و تعدد مستقیم حداقل برابر با $|Z|-1)/2$ است.

تقسیم قالب معکوس[5] $B_b(Z)$ و تعدد معکوس $m_b(Z,r)$ برای رشته Z به‌طور مشابه تعریف می‌شود با این تفاوت که به‌جای آخرین قالب Z، نخستین قالب در نظر گرفته نمی‌شود. بنابراین تقسیم قالب معکوس تقسیم قالبی با اندیس‌های i با $0 < i \le |Z|_r - 1$ است.

در انتها نیز تعدد یک رشته را بیشینه دو تعدد مستقیم و معکوس تعریف می‌کنیم:

$$m(Z,r) = max\{m_f(Z,r), m_b(Z,r)\}$$

برای حالتی که تنها یک پاسخ وجود داشته باشد، یعنی تنها بازای یک مقدار برای s_0 داشته باشیم $squeeze(s_0, |Z|) = Z$ یک کران امنیتی اثبات می‌شود. اگر $|Z| > b$ باشد، این مسأله محتمل بوده و احتمال وجود بیش از یک پاسخ به‌صورت نمایی با $|Z| - b$ کاهش می‌یابد.

[3] Forward block partition
[4] Forward multiplicity
[5] Backward block partition

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

قضیه: برای $Z = squeeze(s_0, |Z|)$ داده شده، کران بالای احتمال موفقیت در یافتن s بعد از N پرسمان، اگر تنها یک مقدار برای s وجود داشته باشد، برابر است با $m(Z,r)\frac{N}{2^c}$.

اثبات: فرض کنید $F_1(Z)$ مجموعه‌ای از جایگشت‌ها برای f باشد به‌گونه‌ای که تنها یک پاسخ برای مسأله بازیابی حالت داخلی (برای مثال بازیابی حالت داخلی متناظر با Z) باشد. برای یک مقدار داده شده برای s در $F_1(Z)$، از آنجا که در مسأله محدودیتی برای بخش درونی بیان نشده است، بخش درونی $f(s)$ یا $f^{-1}(s)$ می‌تواند به‌صورت متقارن از بین 2^c مقدار ممکن انتخاب شود. به بیان دیگر، اگر $\widehat{s_0}$ به‌گونه‌ای باشد که $\bar{f}(Z_0||\widehat{s_0}) = Z_1$، آن‌گاه به‌ازای هر $\widehat{s'_0} \neq \widehat{s_0}$ جایگشت دیگر $f' \in F_1(Z)$ وجود دارد، به‌صورتی‌که $\bar{f}'(Z_0||\widehat{s'_0}) = Z_1$. این تقارن برای چند مقدار درونی، به‌صورت مستقل از یکدیگر، مادامی‌که مقادیر خروجی متفاوت هستند نیز وجود دارد. برای مثال، اگر $Z_1 \neq Z_2$ و $(\widehat{s_1}, \widehat{s_2})$ به‌گونه‌ای باشند که مقدار بیرونی $f(s_i)$ برابر Z_i برای $i = 1,2$ باشد، آن‌گاه برای هر $(\widehat{s_1}, \widehat{s_2}) \neq (\widehat{s'_1}, \widehat{s'_2})$ جایگشت دیگر $f' \in F_1(Z)$ وجود دارد که $(\widehat{s'_1}, \widehat{s'_2})$ در همان تساوی صدق می‌کند.

ابتدا وضعیت $|Z|_r = 2$ را بررسی می‌کنیم. در این‌صورت $m(Z,r) = 2$ است. فرض کنید $F_1(Z, a_0, a_1)$ زیرمجموعه‌ای از $F_1(Z)$ باشد که در آن a_0 پاسخی برای $\widehat{s_0}$ بوده و $f(Z_0||a_0) = (Z_1||a_1)$. مجموعه $F_1(Z, a_0, a_1)$ مجموعه $F_1(Z)$ را به 2^{2c} زیرمجموعه با اندازه‌های یکسان که با a_0 و a_1 مشخص می‌شوند، افراز می‌کند. و یا به بیان دیگر، a_0 و a_1 مجموعه را به‌صورت متعامد تقسیم می‌کنند.

هدف مهاجم تعیین آن است که جایگشت f در کدام زیرمجموعه $F_1(Z, a_0, a_1)$ قرار دارد. برای این منظور، مهاجم دو نوع پرسمان صورت می‌دهد:

- پرسمان مستقیم: برای حدس a، پرسمان $f(Z_0||a)$ را انجام داده و تطابق بخش خروجی پاسخ با Z_1 را بررسی می‌کند.
- پرسمان وارون: برای حدس a، پرسمان $f^{-1}(Z_1||a)$ را انجام داده و تطابق بخش خروجی پاسخ با Z_0 را بررسی می‌کند.

از آنجا که زیرمجموعه‌های $F_1(Z, a_0, a_1)$ مجموعه $F_1(Z)$ را به‌صورت متعامد با a_0 و a_1 تقسیم می‌کنند، پرسمان مستقیم، بدون کاهش مجموعه مقادیر ممکن برای a_1، تعیین می‌کند که آیا a_0 پاسخ مورد نظر هست یا خیر؛ و همین‌طور در جهت مخالف برای پرسمان‌های وارون. بنابراین، بعد از N_f پرسمان مستقیم و N_b پرسمان وارون، احتمال موفقیت برابر است با

$$1 - (1 - N_f 2^{-c})(1 - N_b 2^{-c}) \leq N 2^{-c}$$

که در آن احتمال برای تمامی جایگشت‌های f که به‌صورت یکنواخت از $F_1(Z)$ برداشته شده محاسبه شده است.

حال وضعیت عمومی را که در آن $|Z|_r \geq 2$ است، بررسی می‌کنیم. اگر Z_iها متفاوت باشند، و یا به بیان دقیق‌تر، اگر $m(Z,r) = 1$، استدلالی مشابه آنچه بیان شد، خواهیم داشت. در غیر این‌صورت، برای درنظرگرفتن امکان تکرار Z_iها تغییراتی باید اعمال شود. برای یک مجموعه مشخص‌شده برای اندیس‌های $i, k, \dots$ در یک زیرمجموعه $B_{(j)}$، ممکن است، محدودیت برای مقادیر ممکن برای مقادیر درونی $\widehat{s_i}$ وجود داشته باشد. برای نمونه، اگر $Z_{i-1} \neq Z_{k-1}$ و یا $Z_{i+1} \neq Z_{i_k+1}$، آن‌گاه به‌حتم $\widehat{s_i} \neq \widehat{s_k}$. در نمونه‌ای دیگر، Z می‌تواند متناوب بوده و مقادیر s_i یکسان باشند.

مهاجم می‌تواند مقداری برای قسمت درونی $\widehat{s_i}$ برای تمام $i \in B_{(j)}$ها را در یک پرسمان تنها به‌صورت زیر حدس بزند. مهاجم برای حدس a، یک پرسمان مستقیم برای بررسی تطابق مقدار بیرونی خروجی $f(Z_{(j)}||a)$ با Z_{i+1} برای هر $i \in T_{(j)}$ صورت می‌دهد. استدلالی مشابه برای پرسمان‌های وارون نیز قابل ارائه است. درواقع، مهاجم پرسمان وارون را برای بررسی تطابق مقدار بیرونی خروجی $f^{-1}(Z_{(j)}||a)$ با Z_{i-1} برای هر $i \in B_{(j)}$ صورت می‌دهد. بنابراین، شانس یک پرسمان مستقیم (به‌صورت متناظر وارون) برای تطابق مورد نظر با مقدار $m_f(Z,r)$ (به‌صورت متناظر با $m_b(Z,r)$) محدود می‌شود. اگر بخش بیرونی خروجی $f(Z_{(j)}||a)$ مقدار Z_{i+1} را برای بعضی از $i \in T_{(j)}$ها نتیجه دهد، مهاجم می‌تواند با پرسمان اضافی $f\big(f(Z_{(j)}||a)\big)$ و بررسی تطابق مقدار بیرونی خروجی آن با Z_{i+2} و به همین ترتیب حدس مورد نظر را بررسی کند. در کران بالای معرفی‌شده، این پرسمان‌های اضافی درنظر گرفته نشده است. اگر r بزرگ باشد، تنها یک پرسمان اضافی به‌ازای هر $2^r/m_f(Z,r)$ حدس نیاز بوده و کران معرفی‌شده از دقت خوبی برخوردار خواهد بود. اگر r=1 باشد، این پرسمان‌های اضافی تأثیر قابل توجهی داشته و کران معرفی‌شده دقت پایین‌تری خواهد داشت.

فرض کنید $F_1(Z, a_0, a_1, \dots, a_{|Z|_r-1})$ زیرمجموعه‌ای از $F_1(Z)$ باشد که $(a_0, a_1, \dots, a_{|Z|_r-1})$ پاسخ $(\widehat{s_0}, \widehat{s_1}, \dots, \widehat{s_{|Z|_r-1}})$ است. در حالت کلی، $(a_0, a_1, \dots, a_{|Z|_r-1})$ مجموعه $F_1(Z)$ را به‌صورت متعامد تقسیم نمی‌کنند. حال $|B|$ بردار $A_{(j)}$ که هر بردار شامل $|B_{(j)}|$ المان a_i و $i \in T_{(j)}$ است. از آنجا که بردارهای $A_{(j)}$

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

محدودیت متفاوت بودن بخش بیرونی مقادیر خروجی f را به همراه دارند، مجموعه $F_1(Z)$ را به‌صورت متعامد تقسیم می‌کند؛ بنابراین بعد از n حدس، احتمال یافتن یک پاسخ بیشینه برابر $m(Z,r)2^{-c}n$ است، که در آن احتمال برای تمامی جایگشت‌های f که به‌صورت یکنواخت از $F_1(Z)$ انتخاب شده محاسبه شده است. کران معرفی‌شده در قضیه از این حقیقت که تعداد پرسمان‌های متوسط لازم برای هر حدس غلط کمتر از $1/(1-2^{-r})$ نیست، نتیجه شده است.

مهاجم فعال

در بعضی از مدهای اسفنج، مانند مدهای وابسته با ساختار دوتایی، مهاجم می‌تواند قالب‌های ورودی را جذب کند. این مسأله در قضیه بعدی پوشش داده شده است. فرض می‌کنیم مهاجم می‌تواند قالب‌های P_i را که در هر تکرار وارد می‌شود انتخاب کند، بدین معنی که مد $Z_{i+1} = f\left(s_i \oplus (P_i||0^c)\right) = s_{i+1}$ را محاسبه و مهاجم $\overline{s_{i+1}}$ را مشاهده می‌کند. پاسخ یک مسأله نمونه با قالب‌های ورودی به‌صورت $P = (P_0, P_1, \dots, P_m)$ تعیین می‌شود (قالب آخر P_m مهم نبوده و تنها برای ساده‌سازی نمادگذاری آورده شده است).

قضیه: برای یک مسأله بازیابی حالت داخلی Z و P، و مشخص‌بودن اینکه فقط و فقط یک پاسخ $\hat{s_0}$ برای این مسأله وجود دارد، احتمال موفقیت بعد از N پرسمان بیشینه برابر است با:

$$max\{m_f(Z \oplus P, r), m_b(Z,r)\}\frac{N}{2^c}$$

اثبات: استدلال برای اثبات این قضیه نیز مشابه قضیه قبل است، با این تفاوت که پرسمان‌ها اندکی متفاوت هستند:

- در پرسمان مستقیم، مهاجم برای حدس a تساوی $\bar{f}\left((Z_i \oplus P_i)||a\right) = Z_{i+1}$ را بررسی می‌کند.
- در پرسمان معکوس، مهاجم برای حدس a تساوی $\bar{f}\left((Z_{i+1})||a\right) = Z_i \oplus P_i$ را بررسی می‌کند.

از آنجا که یک پرسمان مستقیم می‌تواند برای بررسی مقادیر درونی $m_f(Z \oplus P, r)$ اندیس به‌صورت یکجا به‌کار رود، واضح است که لازم است تعدد مستقیم $Z \oplus P$ بجای Z در نظر گرفته شود. توجه کنید مهاجم می‌تواند تعدد مستقیم را، با انتخاب قالب P_i به صورتی‌که $Z_i \oplus P_i$ همواره یک مقدار داشته باشد، بیشینه کند. بدین ترتیب، احتمال موفقیت بعد از N پرسمان برابر خواهد بود با $N2^{-c}(|Z|_r - 1)$.

۵- انقیاد خروجی

هدف پیداکردن حالت داخلی s برای یک رشته داده‌شده Z است به‌طوری‌که $squeeze(s, |Z|) = Z$. این روش تحلیل مشابه تحلیل بازیابی حالت داخلی است، با این تفاوت که رشته داده‌شده Z لزوماً خروجی ساختار اسفنجی مورد بررسی نبوده و یک رشته دلخواه است. در اینجا، تنها رشته‌های Z که از بیش از r بیت تشکیل شده‌اند در نظر گرفته می‌شود. احتمال موفقیت روی تبدیل‌های f (یا جایگشت‌های f) و روی $\hat{s} \in Z_2^c$ با شرط $\overline{f^i(Z_0||\hat{s})} = Z_i, \forall i \in \{1 \dots m\}$، نه‌تنها به طول Z بلکه به ساختار آن نیز وابسته است.

مهاجم می‌تواند به‌صورت تصادفی a را حدس بزند تا زمانی‌که مقداری برای a بیابد؛ به‌طوری‌که $\overline{f(Z_0||\hat{s})} = Z_1$؛ سپس، مقدار $\overline{f^2(Z_0||a)}$ را محاسبه کرده و تساوی آن با Z_2 را بررسی می‌کند. در صورت برقراری تساوی، این روند تا رسیدن به آخرین قالب خروجی ادامه می‌یابد. در صورت عدم برقراری تساوی، مقدار جدیدی برای a حدس و بررسی می‌شود. در هر مرحله، در صورت عدم وجود دور و صرف‌نظرکردن از اریبی‌ها، احتمال صحیح‌بودن قالب بعدی محاسبه‌شده توسط مهاجم برابر با 2^{-r} است. در صورت مواجهه با یک قالب ناصحیح، مهاجم با حدس مقدار دیگری برای a مراحل را تکرار می‌کند. متوسط تعداد دفعات فراخوانی f برای حذف یک حدس به $\frac{1}{1-2^{-r}}$ خیلی نزدیک است.

اگر $|Z| < b$ باشد، احتمال موفقیت یک حدس برابر است با:

$$Pr(success\ with\ guess) = 2^{r-|Z|}.$$

با درنظرداشتن تعداد دفعات فراخوانی f برای یک حدس، برای f یک تبدیل تصادفی و هم‌چنین یک جایگشت تصادفی، تابع هزینه به‌صورت زیر خواهد بود:

$$c_p(output\ binding) \approx \frac{2^{r-1}}{2^{|Z|}}N.$$

زمانی‌که $|Z| > b$، مقدار مورد انتظار برای N از تعداد حالات ممکن برای حالت داخلی بیشتر است. این مسأله نشان می‌دهد مهاجم بایستی بخش بزرگی از مقادیر $\hat{s} \in Z_2^c$ را بررسی کند. از آنجا که بیش از 2^c انتخاب برای $\hat{a}$ وجود ندارد، نتیجه می‌شود که ممکن است، پاسخی وجود نداشته باشد. تنها برای بخشی از تبدیل‌های ممکن (یا جایگشت‌های ممکن) یک مقدار برای قسمت درونی $\hat{s}$ که دنباله خروجی را نتیجه می‌دهد، وجود دارد. احتمال وجود چنین مقداری برای قسمت درونی برای $|Z| > b$ برابر با $\frac{2^{r+c}}{2^{|Z|}}$ است.

۶- نتیجه‌گیری

در این نوشتار به مطالعه حملات عام روی ساختار اسفنجی پرداخته شده است. ایجاد برخورد داخلی، یافتن مسیر به یک حالت داخلی، تشخیص دور در خروجی، بازیابی حالت داخلی و انقیاد خروجی از حملات مهم روی ساختار اسفنجی است که تعیین پیچیدگی آنها با استفاده

[Downloaded from monadi.isc.org.ir on 2026-11-09]
[DOR: 20.1001.1.24763047.1399.9.1.5.8]

از تعریف گراف اسفنج و بهره‌برداری از آن انجام شد. در این مقاله، نحوه اعمال بهینه این حملات و پیچیدگی آنها روی ساختار اسفنجی در مواردی که تابع بروز رسانی یک جایگشت و یا یک تبدیل تصادفی است، نیز تشریح شد. خلاصهٔ نتایج ارائه‌شده در جدول (۱) و جدول (۲) آورده شده است. گفتنی است که مطالعه دقیق پیچیدگی این حملات در زمان انتخاب پارامترها و ارزیابی طرح‌ها از دید طراحی و تحلیل الگوریتم‌های مبتنی بر ساختار اسفنجی می‌تواند ابزاری کارا جهت استفاده از این ساختارها در رمزنگاری ارائه دهد.

(جدول-۱): تابع هزینه برای حملات عمومی

f	برخورد	یافتن مسیر	تشخیص دور	بازیابی حالت داخلی	انقیاد خروجی
تبدیل	$\frac{N(N+1)}{2^{c+1}}$	$\frac{N}{2^c}$	$\frac{N(N+1)}{2^{r+c+1}}$	$\frac{N}{2^c}$	$\frac{1-2^{-r}}{2^{\lvert Z\rvert-r}}N$
جایگشت	$\frac{N(N+1)}{2^{c+1}}-\frac{N(N-1)}{2^{r+c+1}}$	$\frac{N(N+4)}{2^{c+2}}-\frac{N^2}{2^{r+c+2}}$	$\frac{N}{2^{r+c}}$	$\frac{\lvert Z\rvert-1}{2^c}N$	$\frac{1-2^{-r}}{2^{\lvert Z\rvert-r}}N$

(جدول-۲):تابع هزینه ساده شده برای حملات عمومی

f	نرخ	برخورد	یافتن مسیر	تشخیص دور	بازیابی حالت داخلی	انقیاد خروجی
تبدیل	$r \ggg 1$	$2^{-(c+1)}N^2$	$2^{-c}N$	$2^{-(c+r+1)}N^2$	$2^{-c}N$	$2^{-(\lvert Z\rvert-r)}N$
تبدیل	$r=1$	$2^{-(c+1)}N^2$	$2^{-c}N$	$2^{-(c+2)}N^2$	$2^{-c}N$	$2^{-\lvert Z\rvert}N$
جایگشت	$r \ggg 1$	$2^{-(c+1)}N^2$	$2^{-(c+2)}N^2$	$2^{-(c+r)}N$	$(\lvert Z\rvert-1)2^{-c}N$	$2^{-(\lvert Z\rvert-r)}N$
جایگشت	$r=1$	$2^{-(c+2)}N^2$	$2^{-(c+3)}N^2$	$2^{-(c+r)}N$	$(\lvert Z\rvert-1)2^{-c}N$	$2^{-\lvert Z\rvert}N$

۷- مراجع

[۱] Bertoni, G., Daemen, J., Peeters, M., & Assche, G. V. (2011). Cryptographic Sponge Functions. Retrieved from http://sponge.noekeon.org/.

[2] Bertoni, G., Daemen, J., Peeters, M., & Assche, G. V. (2013). Keccak. EUROCRYPT 2013, LNCS 7881, pp. 313-314.

[3] Li, W., Liao, G., Wen, Y., & Gong, Z. (2017). SpongeMPH: A New Multivariate Polynomial Hash Function based on the Sponge Construction. Second International Conference on Data Science in Cyberspace (DSC). Shenzhen: IEEE.

[4] Abdoun, N., El Assad, S., Hammoud, K., Assaf, R., Khalil, M., & Diforges, O. (2018). New Keyed Chaotic Neural Network Hash Function Based on Sponge Construction. International Conference for Internet Technology and Secured Transactions (ICITST). Cambridge: IEEE.

[5] Kumar Singh, P., Monsy, A. V., Garg, R., Dey, S., & Nandi, S. (2019). JSpongeGen: A Pseudo Random Generator for Low Resource Devices. International Conference on Distributed Computing and Internet Technology, ICDCIT 2019, LNCS 11319, (pp. 410-421). Cham: Springer.

[6] AlTawy, R., Rohit, R., He, M., Mandal, K., Yang, G., & Guang, G. (2018). sLiSCP: Simeck-Based Permutations for Lightweight Sponge Cryptographic Primitives. International Conference on Selected Areas in Cryptography. LNCS 10719, pp. 129-150. Cham: Springer.

[7] Bertoni, G., Daemen, J., Assche, G. V., & Peeters, M. (2006). Radiogatun, a Belt-and-Mill Hash Function. Second Cryptographic Hash Workshop, Retrieved from http://radiogatun.noekeon.org/

[8] https://csrc.nist.gov/projects/hash-functions/sha-3-project

[9]https://competitions.cr.yp.to/caesar-submissions.html

[10] NIST. (2015). Secure Hash Standard (SHS) (FIPS 180-4). Retrived fromhttps://csrc. nist.gov/ publications/detail/fips/180/4/final

[11] https://keccak.team/sponge_duplex.html

[12] Chaigneau, C., Fuhr, T., Gilbert, H., Jean, J., Reinhard, J. (2019). Cryptanalysis of NORX v2.0. Journal of Cryptology, 2019, Volume 32, Number 4, pp. 1423-1447.

[۱۳]Reduced KECCAK Using Non-linear Structures. In: Hao F., Ruj S., Sen Gupta S. (eds) Progress in Cryptology- INDOCRYPT 2019. INDOCRYPT 2019. Lecture Notes in Computer Science, vol 11898. Springer, Cham.

[۱۴] Bi, W., Dong, X., Li, Z., Zong, R., Wang, X. (2019). MILP-Aided Cube-Attack-Like Cryptanalysis on Keccak Keyed Modes. Designs Codes and Cryptography, 2019, Volume 87, Issue 6, pp. 1271-1296.

[Downloaded from monadi.isc.org.ir on 2026-11-09]

[DOR: 20.1001.1.24763047.1399.9.1.5.8]