فصلنامه علمی-ترویجی پدافند غیرعامل

سال دوم، شماره ۴، زمستان ۱۳۹۰، (پیاپی ۸): صص ۵۳-۶۴

نقش توابع درهم‌ساز رمزنگاری در امنیت با روی‌کرد پدافند غیرعامل

محمدعلی طاهری[۱]، زین‌العابدین نوروزی[۲]

تاریخ دریافت: ۹۰/۰۷/۲۰
تاریخ پذیرش: ۹۰/۱۰/۲۶

چکیده

امنیت اطلاعات و ارتباطات از اهمیت ویژه‌ای در حوزهای نظامی و امنیتی برخوردار است. ایجاد یک پوشش مناسب و بهینه جهت حفاظت از امنیت و صحت در مقوله‌های ارتباطات و اطلاعات توسط رمزنگاری و پروتکل‌های ارتباطی امکان‌پذیر است. مکانیزم‌هایی که در ارسال و دریافت یک پیام باعث به‌وجود آمدن امنیت می‌گردند پدافند غیرعامل بوده و باعث کاهش آسیب‌پذیری اطلاعات خواهند شد. از طرف دیگر، پروتکل‌های ارتباطی که صحت یک ارتباط و یا اطلاعات را مورد بررسی قرار می‌دهند، باعث کاهش دسترسی افراد غیرمجاز به اطلاعات محرمانه شده و به‌علاوه، آسیب‌پذیری یک شبکه ارتباطی را با فرایندهایی هم‌چون عدم ذخیره‌سازی فایل‌های بزرگ، عدم دسترسی به بخشی از پیام اصلی، به‌وجود آوردن خلاصه پیام با طول بسیار کوچک برای هر پیام سری و غیرسری طولانی و غیره را بسیار کاهش خواهند داد. روند فوق، مفهوم بسیار دقیق از پدافند غیرعامل بوده، زیرا آسیب‌پذیری ارتباطات و اطلاعات را در پروتکل‌های ارتباطی به طور شایسته‌ای کاهش خواهد داد. امنیت یک پیام، توسط رمزنگاری قابل حصول است. در بسیاری از مواقع، هدف ارسال‌کننده و دریافت‌کننده پیام، صحت و سندیت پیام می‌باشد. این مهم توسط پروتکل‌های ارتباطی هم‌چون امضاهای رقمی، توابع درهم‌ساز، کدهای اعتباری پیام و غیره امکان‌پذیر است. توابع درهم‌ساز، نقش اساسی را در پروتکل‌های ارتباطی به‌عهده دارند؛ به‌عنوان نمونه، عدم استفاده از تابع درهم‌ساز در یک امضای رقمی باعث عدم کارایی امضا خواهد شد. در این مقاله، مفهوم دقیق امنیت و کاربردهای توابع درهم‌ساز را بیان داشته و سپس به سیر تحول این توابع اشاره می‌کنیم. در ادامه، یکی از توابع جدید به‌نام گروستال (یکی از پنج تابع درهم‌سازی است که به مرحله سوم مسابقه *NIST* رسیده و ما فکر می‌کنیم که این تابع در نهایت در سال ۲۰۱۲ توسط این مؤسسه به‌عنوان تابع درهم‌ساز جهانی انتخاب خواهد شد) را از نظر ساختاری مورد بررسی قرار داده و به‌عنوان کار جدید، مقاوم بودن این تابع در مقابل حملات مکعبی را مورد تحلیل قرار می‌دهیم.

کلیدواژه‌ها: توابع درهم‌ساز، کدهای اعتباری پیام، رمزنگاری و حملات مکعبی

۱- دانشجوی کارشناسی ارشد مخابرات- رمز- دانشگاه امام حسین(ع)، Email: Taheri.nodh@yahoo.com - نویسنده مسئول

۲- مدرس و عضو هیئت علمی دانشگاه جامع امام حسین(ع) گروه ریاضی-رمز، Email: Znorozi@ihu.ac.ir

مقدمه

قرن بیستم عصر جمع‌آوری، پردازش، توزیع اطلاعات و حفظ امنیت این اطلاعات است. می‌توان مهم‌ترین اتفاق قرن بیستم را پیوند علم رایانه با مخابرات دانست که منجر به تحولات عظیم دو صنعت شد. در نتیجهٔ این پیوند، حفظ امنیت اطلاعات از اهمیتی ویژه برخوردار شد. با پیدایش شبکه‌های رایانه‌ای و نقش آن‌ها در مخابرات، در واقع امنیت اطلاعات، جایگزین امنیت شبکه‌های رایانه‌ای گردید.

امنیت، حصول اطمینان از عدم دسترسی افراد غیرمجاز به پیام‌های محرمانه و جلوگیری از دست‌کاری در این پیام‌ها می‌باشد. مشکلات امنیت ارتباطات و اطلاعات به‌طور کلی به چهار رده نزدیک و مرتبط به‌هم تقسیم‌بندی می‌شوند:

۱- **سری ماندن اطلاعات**[1]: این مقوله، متضمن انجام عملیاتی است که اطلاعات را از دسترس کاربران غیر مجاز و بیگانه دور نگاه می‌دارد.

۲- **احراز هویت کاربران**[2]: این فرایند باعث تأیید هویت طرف مقابل ارتباط، قبل از آن‌که اطلاعات حساس در اختیار او قرار گیرد یا در معاملات تجاری شرکت داده شود، می‌گردد،

۳- **غیر قابل انکار بودن پیام‌ها**[3]: این مهم با امضاهای رقمی سر و کار دارد و به اطلاعات و مستندات، هویت حقوقی می‌دهد.

۴- **نظارت بر صحت اطلاعات**[4]: چگونگی اطمینان از پیام دریافتی - که اصیل بوده و در حین انتقال، در آن دست‌کاری و تحریف صورت نگرفته است - برعهده این فرایند است.

در ادامه، مفهوم رمزنگاری[۱] را بیان و سپس به مفهوم دقیق توابع درهم‌ساز اشاره خواهیم نمود، که این توابع نقش اساسی و بسیار حیاتی را برای برخی از پروتکل‌های ارتباطی بازی خواهند کرد.

ساختار مقاله به شرح زیر است: در فصل اول، مفهوم دقیق رمزنگاری بیان شده؛ در فصل دوم تعریف توابع درهم‌ساز را ارائه داده و در ادامه به امنیت این توابع اشاره می‌کنیم. مقدمه‌ای در خصوص و چگونگی مسابقه SHA-۳ را در فصل سوم بیان نموده و ساختار تابع درهم‌ساز گروستال را در فصل چهارم آورده‌ایم. در فصل پنجم حملات مکعبی را بیان داشته و حمله مکعبی به تابع گروستال را در فصل ششم و در فصل پایانی، نتیجه‌گیری را ارائه نموده‌ایم.

۱- رمزنگاری

رمزنگاری به معنای محرمانه نوشتن متون است. پیامی که باید رمزنگاری شود، متن آشکار[5] نامیده می‌شود و توسط یک تابع خاص با پارامتری به نام کلید[6] به متن رمزی[7] تبدیل می‌گردد. این متن بر روی کانال ناامن منتقل خواهد شد. هنر شکستن رمز بدون در اختیار داشتن کلید آن، «علم تحلیل رمز»[8] نام دارد. به‌طور کلی، روند تکامل رمزنگاری را می‌توان به چهار مرحله زیر تقسیم کرد:

- **مرحله اول**: استفاده از سیستم‌های ساده جانشینی و جابه‌جایی برای رمزنگاری؛ در این مرحله، بیشتر قلم و کاغذ و ماشین‌های ساده مکانیکی مورد استفاده قرار می‌گرفتند.
- **مرحله دوم**: از ابتدای قرن بیستم تا دهه ۱۹۵۰؛ در این مرحله، از وسایل پیچیده مکانیکی و الکترومکانیکی استفاده شده و به تبع آن سیستم‌های رمزنگاری پیچیده‌تری ابداع گردید.
- **مرحله سوم**: این مرحله با انتشار مقاله بسیار مهم شانون در سال‌های ۱۹۴۸ و ۱۹۴۹ و پیشرفت سریع در صنایع میکروالکترونیک در دهه ۱۹۶۰ شروع شد و هنر رمزنگاری به علم رمزنگاری مبدل و به «دوره رمزهای متقارن» معروف گردید.
- مرحله چهارم: از اواخر دهه ۱۹۷۰ با پیشنهاد سیستم‌های رمزنگاری با کلید عمومی توسط دیفی و هلمن شروع شد.

دو اصل اساسی زیر باید در تمام سیستم‌های رمزنگاری رعایت شود:

۱- **افزونگی:** تمام پیام‌های رمزشده باید شامل مقداری افزونگی باشند؛ به‌عبارت دیگر لزومی ندارد که اطلاعات واقعی به همان‌گونه که هستند رمز و ارسال شوند.

۲- **تازگی پیام‌ها:** در رمزنگاری، عملیاتی جهت اطمینان از جدید بودن پیام دریافتی لازم است. این فرایند برای جلوگیری از ارسال مجدد پیام‌های قدیمی توسط یک مهاجم فعال الزامی است.

با توجه به اهداف مهاجم می‌توان دو نوع حمله را برای سیستم‌های رمزنگاری در نظر گرفت:

۱- **حمله غیرفعال**[9]: حمله‌ای است که معمولا توسط استراق‌سمع انجام می‌گیرد و مهاجم هیچ‌گونه دخل و تصرفی در اطلاعات ارسالی ندارد.

۲- **حمله فعال**[10]: حمله‌ای است که دشمن تلاش می‌کند اطلاعات ارسالی را تغییر داده و اطلاعات جدیدی را وارد یا از سیستم خارج کند. در این‌جا مهاجم از ابتدا تا انتهای ارتباط بین فرستنده و گیرنده پیام، می‌تواند فعال باشد.

با توجه به دو نوع حمله اشاره شده فوق، در سیستم‌های رمزنگاری همواره با یکی از دو مسئله زیر روبه‌رو هستیم:

1- Secracy
2- Authentication
3- Nonrepudiation
4- Integrity Control
5- Plaintext
6- Key
7- Ciphertext
8- Cryptoanalysis
9- Passive attack
10- Active attack

۱- **محرمانه ماندن پیام**[1]: باید سیستم رمزنگاری به‌گونه‌ای طراحی گردد که دشمن نتواند با استفاده از امکانات موجود، کوچک‌ترین اطلاعاتی را در مورد پیام ارسالی به‌دست آورد.

۲- **معتبر ماندن پیام**[2]: در این مورد باید سیستم رمزنگاری به‌گونه‌ای طراحی شود که دشمن نتواند اطلاعات را تغییر دهد و یا اطلاعات جدیدی را وارد سیستم نموده و یا اطلاعات قبلی را تکرار کند.

محرمانگی پیام ایجاب می‌کند که دشمن بر اساس متن رمزشده دریافتی، نتواند متن اصلی را به‌دست آورد. اعتبار پیام ایجاب می‌کند که دشمن نتواند یک متن جعلی رمزشده را به‌جای متن معتبر رمزشده قرار دهد. به‌عبارت دیگر، اگر دشمن چنین کاری انجام داد، گیرنده بتواند جعل صورت‌گرفته در پیام را کشف کند.

در یک نگاه کلی، سیستم‌های رمزنگاری را می‌توان به دو سیستم متقارن[3] و نامتقارن تقسیم نمود.

- **سیستم‌های متقارن** عبارت است از سیستم‌هایی که در آن‌ها کلید رمزنگاری و رمزگشایی یکسان بوده و یا به‌راحتی بتوان آن‌ها را با اطلاع از کلید یکدیگر به‌دست آورد.
- **سیستم‌های نامتقارن** عبارت است از سیستم‌هایی که در آن‌ها کلیدهای رمزگذاری و رمزگشایی متفاوت بوده و می‌بایستی محاسبه یک کلید از روی دیگری از نظر محاسباتی (در زمان چندجمله‌ای) غیرممکن باشد. بنابراین، یکی از کلیدها می‌تواند آشکار باشد (کلید عمومی)، مشروط بر آن‌که اطلاعات یا سرنخی در مورد کلید دیگر (کلید خصوصی) به دشمن ندهد.

همان‌طورکه اشاره شد، امنیت یک پیام توسط علم رمزنگاری تأمین می‌گردد. در بسیاری از مواقع، امنیت همراه با سندیت یک پیام مد نظر می‌باشد. یکی از راه‌کارهای بسیار مهم در سندیت و صحت یک پیام، استفاده از توابع درهم‌ساز رمزنگاری[4] است که نقش مهمی در رمزنگاری دارند. زیرا اعتبار پیام با استفاده از این نوع توابع قابل تضمین است. در ادامه، به این نوع توابع اشاره و کاربردهایی از توابع درهم‌ساز را بیان خواهیم داشت.

۲- توابع درهم‌ساز

یک تابع درهم‌ساز، تابعی مانند h می‌باشد که ورودی‌های به طول دلخواه را به خروجی‌های با طول ثابت در زمان چندجمله‌ای نگاشت می‌کند که به این خروجی «چکیده پیام» گفته می‌شود[۱]. این نگاشت باید به‌گونه‌ای باشد که خروجی به‌طور موثری یکتا باشد، یعنی در حالی‌که ممکن است ورودی‌های دیگر نیز همان خروجی را تولید کنند وجود داشته باشند؛ ولی احتمال پیدا کردن یک چنین ورودی‌هایی باید ناچیز باشد، به‌طوری که از لحاظ عملی پیدا کردن آن‌ها سخت باشد. به‌علاوه، این نگاشت باید معکوس‌پذیر نباشد. بنابراین، تابع درهم‌ساز $h:\{0,1\}^* \rightarrow \{0,1\}^m$ تابعی است با حداقل دو خاصیت زیر:

- خروجی h باید به‌طور موثری یکتا باشد. بدین مفهوم که برای تولید دو پیام متفاوت x و x' با تساوی $h(x)=h(x')$ نیازمند به $2^{\frac{m}{2}}$ عمل درهم‌سازی باشد.
- تابع h باید معکوس‌پذیر نباشد، یعنی برای خلاصه پیام y، طوری که $h(x)=y$، محاسبه x از طریق y باید حداقل نیازمند 2^m عمل درهم‌سازی باشد.

به‌طور رسمی، یک تابع درهم‌ساز به‌صورت زیر تعریف می‌شود:

تعریف: چهارتایی (X,Y,H,K) را یک خانواده درهم‌ساز گوییم، که در آن X مجموعه تمام پیام‌های ممکن، Y مجموعه تمام خلاصه‌پیام‌ها[5]، K مجموعه تمام کلیدهای ممکن و H خانواده توابع اعمال‌شده به پیام باشد؛ به‌گونه‌ای که به ازای هر $k \in K$ و $h_k \in H$ رابطه زیر برقرار باشد:

$$h_k: X \rightarrow Y; h_k(x)=y, \forall x \in X,\ y \in Y, \forall k \in K$$

در این صورت، H مجموعه توابع درهم‌ساز و h_k، تابعی درهم‌ساز از H می‌باشد[۲۴].

۲-۱- امنیت توابع درهم‌ساز

یک تابع درهم‌ساز را امن گوییم هرگاه در مقابل مسائل زیر مقاوم باشد:

- ***پیش‌تصویر***[6]: در این حمله، مهاجم، تابع درهم‌ساز h و خلاصه پیام y را در اختیار دارد و سعی می‌کند پیام x را طوری پیدا کند که $h(x)=y$.
- ***پیش‌تصویر دوم***[7]: در این حمله، مهاجم، تابع درهم‌ساز h و پیام x را در اختیار دارد و سعی می‌کند پیام دیگری x' را طوری پیدا کند که $h(x)=h(x')$.
- ***برخورد***[8]: در این حمله، مهاجم، تابع درهم‌ساز h را در اختیار دارد و سعی می‌کند دو پیام متفاوت x و x' را طوری پیدا کند که $h(x)=h(x')$.
- ***پیش‌تصویر جزیی***[9]: برای هر تابع درهم‌ساز h، و چکیده داده شده مانند y، طوری‌که $h(x)=y$ باید پیدا کردن چند بیت از پیام x به‌همان سختی پیدا کردن کل پیام باشد.

1- Privacy problem
2- Authentication problem
3- Symmetric
4- Cryptographic Hash Functions
5- Message digest
6- Preimage
7- Second preimage
8- Collision
9- Pseudo preimage

علاوه بر این، اگر از کل پیام x فقط t بیت آن را ندانیم، برای پیدا کردن این t بیت، باید نیازمند 2^{t-1} عمل درهم‌سازی باشیم.

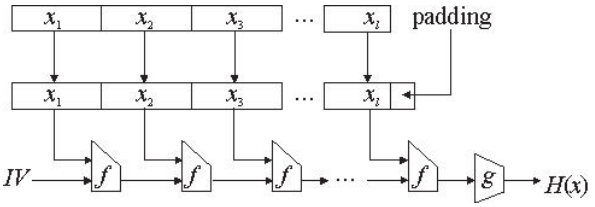
یکی از مسائل مهم در امنیت توابع درهم‌ساز، نوع طراحی این توابع می‌باشد. یکی از طراحی‌های بسیار مناسب، طراحی بر مبنای رمز بلوکی بوده که همزمان قادر به تامین امنیت و سرعت مناسب بوده است. یکی از بهینه‌ترین این نوع طرح‌ها، طرح زنجیره‌ای مرکل-دمگارد است که در ادامه این طرح به‌صورت خلاصه بیان خواهد شد.

۲-۲- طرح زنجیره مرکل-دمگارد

بسیاری از توابع درهم‌ساز مانند $MD5$ [۲] و $SHA-1$ [۳] بر مبنای ساختار زنجیره مرکل-دمگارد[۴و۵] طراحی شده‌اند. در این طرح، یک تابع درهم‌ساز مانند h با استفاده از یک تابع فشرده‌ساز مانند f ساخته می‌شود و می‌توان ثابت کرد که اگر تابع فشرده‌ساز f مقاوم در برابر برخورد باشد، آنگاه تابع درهم‌ساز h نیز مقاوم در مقابل برخورد است. در این ساختار، تابع فشرده‌ساز استفاده $f: \{0,1\}^m \times \{0,1\}^b \to \{0,1\}^m \quad m.b \geq 1$ می‌شود. روش کار به این صورت است که پیام‌ها به داخل t بلوک b بیتی از X_1 تا X_t تقسیم می‌شوند. اگر تعداد همه بیت‌ها مضرب صحیحی از طول بلوک یعنی b نباشد، یک روش لایه‌گذاری[1] مشخص می‌شود. یک روش لایه‌گذاری روی یک پیام مانند x، به این صورت است که ابتدا یک بیت ۱ به انتهای پیام ورودی اضافه و سپس به تعداد کافی بیت صفر به پیام الحاق[2] شده تا طول حاصل در پیمانه b، برابر با ۴۴۸ بیت گردد. پیام جدید را x' در نظر گرفته و در انتها نمایش دودویی طول x، به‌صورت یک بردار ۶۴ بیتی در نظر گرفته شده و به انتهای بردار x' الحاق می‌شود تا پیام لایه‌گذاری شده نهایی تولید شود. طول این پیام لایه‌گذاری شده، مضرب صحیحی از b خواهد بود. تابع درهم‌ساز h با تابع فشرده‌ساز یا تابع دور f می‌تواند به‌صورت زیر تعریف شود:

$$H_0 = IV$$
$$H_i = f(X_i, H_{i-1}) \quad i = 1,2,\ldots,t$$
$$h(X) = H_t$$

در تعریف فوق، H_iها متغیرهای میانی یا متغیرهای زنجیره‌ای هستند که طول آن‌ها n بیت می‌باشد و X_i ها بلوک‌های b بیتی هستند. نتیجه تابع درهم‌ساز با $h(X)$ نمایش داده می‌شود که در آن IV مقدار اولیه است. یک شمای کلی از چنین طرحی در شکل (۱) ارائه شده است.



شکل ۱- نمای کلی از یک تابع درهم‌ساز تکرار شونده[۲۱]

۲-۳- کاربردهای توابع درهم‌ساز

در این بخش به برخی از کاربردهای مهم و اساسی توابع درهم‌ساز اشاره می‌کنیم.

۲-۳-۱- امضای رقمی

در این بخش قصد داریم نگاهی گذرا به امضای رقمی[۶] داشته باشیم و بعد نتیجه‌گیری کنیم که چرا باید از توابع درهم‌ساز در این طرح‌ها استفاده کنیم و بررسی کنیم که چرا امنیت امضای رقمی تا حد زیادی به امنیت تابع درهم‌ساز استفاده‌شده در آن وابسته است. در این‌جا به سیستمی نیاز است که بر اساس آن، فرستنده، پیام امضا شده را برای گیرنده پیام بفرستد به‌گونه‌ای که شرایط زیر به‌درستی احراز شوند:

۱- گیرنده بتواند هویت شخص فرستنده پیام را بررسی کند،

۲- فرستنده بعدا نتواند محتوای پیام ارسالی خود را انکار کند،

۳- گیرنده نتواند پیام‌های جعلی برای خود بسازد و ارسال آن‌ها را به دیگران نسبت بدهد.

در عمل، طول اکثر پیام‌هایی که می‌خواهیم امضا کنیم از طول پیام بزرگ‌تر است. حال سوال این است که چگونه برای متن‌های طولانی به‌طور موثری امضا را تولید کنیم. شاید اولین روشی که به ذهن هرکسی خطور کند این است که با روشی مشابه با رمزهای بلوکی از مدها استفاده کنیم، یعنی پیام x به بلوک‌هایی برابر با طول ورودی الگوریتم طرح امضا تقسیم شود و هر بلوک را به‌طور جداگانه امضا کنیم. این روش باعث بروز دو مشکل می‌شود: بار محاسباتی بالا و محدودیت‌های امنیتی. لذا نیازمند روشی هستیم که اولا بار محاسباتی زیادی نداشته باشد، و ثانیاً از امنیت کافی برخوردار باشد. بهترین روش برای حل این مشکلات، استفاده از توابع درهم‌ساز است.

۲-۳-۲- اعتبار اطلاعات

ایده اصلی استفاده از توابع درهم‌ساز رمزنگاری، در واقع کاهش دادن حفاظت از اعتبار اطلاعات[۶] با طول دلخواه به حفاظت از محرمانگی و یا اعتبار با طول ثابتی از اطلاعات می‌باشد. در این ایده دو پرسش وجود دارد: اولا چه وجه تمایزی وجود دارد؟ بین حفاظت از اعتبار اطلاعات همراه با محرمانگی و بدون محرمانگی. ثانیا این‌که آیا حفاظت از اعتبار یک پیام، نیاز به محرمانگی کلید دارد یا وابستگی به

1- Padding
2- Concatenate

مقدار تابع درهم‌ساز کافی است؟. در ادامه بـه ایـن موضـوع خـواهیم پرداخت.

الف- اعتبار بدون محرمانگی: اگر در انتقال یک پیام فقـط اعتبـار مد نظر باشد، استفاده از کدهای اعتباری پیام یا MAC[1] ساده‌تـرین راه است. به منظور حفاظت از اعتبار اطلاعات، کـافی اسـت کـه کـد اعتبار پیام اصلی را محاسبه کنیم و این مقـدار محاسـبه شـده را بـه اطلاعات الحاق کنیم. حال سندیت اطلاعات، وابسته به محرمـانگی و اعتبار کلید مخفی است. بنابراین در این‌جـا مـسئله مـدیریت کلیـد، نقش اساسی پیدا می‌نماید. روش دیگر، استفاده از MDC[2] می‌باشد. در این روند اعتبار اطلاعات تبدیل به اعتبار یک رشته پیـام بـا طـول ثابت می‌گردد. مزیت این روش نسبت به روش قبل، این است کـه در این روش نیاز به مدیریت کلید نیست.

ب - اعتبـار همـراه بـا محرمـانگی: اگـر بـرای پیـامی اعتبـار و محرمانگی با هم مد نظـر باشـد در ایـن صـورت اسـتفاده از کـدهای اعتباری پیام و رمزنگاری به‌صورت توام، راه‌کار است. اشـکال اساسـی در ایـن روش، مـدیریت کلیـد دوگانـه اسـت. روش دوم اسـتفاده از MDC خواهد بود که در این روش، اشکال قبلی مرتفع می‌گردد.

۲-۳-۳- کاربرد توابع درهم‌ساز در بررسـی عـدم مخـدوش شدن فایل‌های بزرگ

یکی دیگر از کاربردهای مهم توابع درهم‌ساز می‌تواند حصول اطمینان از عدم مخدوش شدن یک فایل ذخیره‌شده باشد؛ خصوصاً در حـالتی که حجم فایل خیلی بزرگ است. به‌عنوان مثال، فرض کنید یک فایل با حجم خیلی بزرگ را در رایانه ذخیره کرده‌ایم و در عین حال کپـی مطمئنی را نیز در جای دیگر در اختیار داریـم. حـال در هـر بـار کـه می‌خواهیم از فایل ذخیره شده در رایانه خـود اسـتفاده کنـیم، بایـد مطمئن شویم که این فایل بر اثر عوامل مختلف (مثل ویـروس‌هـای رایانه‌ای) مخدوش نشده باشـد. یـک روش مفیـد و بهینـه مـی‌توانـد استفاده از تابع درهم‌ساز باشد؛ به این صورت که تابع درهم‌ساز را بـه فایل خودمان اعمال می‌کنیم و چکیـده حاصـل را بـا چکیـده فایـل مطمئن که حاصل اثر همان تابع درهم‌ساز بر آن فایل مطمئن اسـت مقایسه می‌نماییم؛ در صورت تطابق، فایـل سـالم اسـت و در صـورت عدم تطابق چکیده‌ها، به احتمال بسیار زیاد فایـل اجرایـی مخـدوش شده است و باید فایل مورد اطمینان را دوباره کپی نماییم؛ این از نظر راندمان، نسبت به هربار کپی کردن، بهتر و مطلوب‌تر است. خـصوصا وقتی که حجم فایل خیلی بزرگ باشد.

۲-۳-۴- کاربردهای دیگر توابع درهم‌ساز

توابع درهم‌ساز در سایر زمینه‌های رمزنگاری کاربرد دارنـد. بـه‌عنـوان نمونه، توابع درهم‌ساز می‌توانند در سیستم‌های دانایی صفر[3] نیز مورد استفاده واقع شوند و راندمان را بالا ببرنـد. همچنـین در کاربردهـایی که اعتبار بسیار حائز اهمیت است، ایـن توابـع نقـش حیـاتی را ایفـا می‌کنند. نمونه‌هایی از این کاربردها پروتکل‌هـای مختلـف اینترنـت[4] (IP)، پروتکل‌های امنیـت وب از قبیـل SET[5]، SSL[6]، سیـستم‌هـای پرداخت الکترونیکی[7]، سیستم‌های چند منظوره و توسعه‌یافته پـست الکترونیکی امن مانند S/MLME[8] و PGP[9]، برنامه‌های کاربردی احراز هویت مانند PKI[10]، تجارت الکترونیکی، شبکه‌های سراسری بـانکی، کارت‌های هوشمند و غیره می‌باشند [۱۰-۷].

به‌علت نقش مهم توابع در نوع کاربرد، دو مقوله مهم در این نوع توابع عبارت‌اند از امنیت و سرعت. بنابراین در طراحی یک تابع درهـم‌سـاز می‌بایست به دو نکته فوق توجه دقیق نمود به‌طوری‌که:

۱- جزیی‌ترین تغییر در ورودی یک پیام باعث تغییر در تمام خلاصـه پیام گردد،

۲- از سرعت قابل قبولی برخوردار باشد.

در ادامه با توجه به نکات فوق، چگونگی شکل‌گیـری ایـن توابـع را از نظر ساختاری مورد اشاره قرار خواهیم داد.

۳- مسابقه SHA − ۳

دو خانواده از توابع درهم‌ساز که بسیار مورد توجـه قـرار گرفتـه و در حال حاضر نیز یکی از مولفه‌های مهم در پروتکل‌های ارتباطی هستند عبارت‌اند از:

۱- خانواده MDها؛ شامل MD۴، MD۵، RIPEMD

۲- خـــانواده SHAهـــا؛ شـــامل SHA − ۱، SHA − ۲۵۶، SHA − ۳۸۴، SHA − ۵۱۲

تابع درهم‌ساز MD۴ در سـال ۱۹۹۰ توسـط رایوسـت[11] ارائـه و در سال ۱۹۹۲ یک نسخه کامل‌تر از آن یعنی MD۵ را پیشنهاد داد. در سال ۱۹۹۶ یک نقطه‌ضعف در فشرده‌ساز این تابع پیدا شد، هر چنـد این نقطه‌ضعف تهدیدی برای کل تابع درهم‌ساز نبود، اما دلیلی موجه برای این‌که نگاه‌ها بسوی استفاده از SHA − ۱ معطوف گردد شد. ولی در عین حال باز هم یکی از توابع درهم‌ساز پرکاربرد بـاقی مانـد. در سال ۲۰۰۴ یـک گـروه از محققـین چینـی بـه سرپرسـتی خـانم وانـگ[12] و همکـارانش حملـه مـوثری بـر روی MD۵ انجـام دادنـد

1- Message authentication cods
2- Message digest code

3- Zero Knowledge
4- Internet Protocol
5- Secure Electronic Transaction
6- Secure Socket Layer
7- Electronic Payment System
8- Secure Multipurpose Internet Mail Extension
9- Pretty Good Privacy
10- Public Key Infrastructure
11- Rivest
12- Wang

به‌طوری که باعث شکست خانواده MDها و SHA-۰ شدند[۱۱]. آن‌ها با تغییر جزئی در حمله تفاضلی کلاسیک توانستند برخوردهایی با ورودی‌های دو بلوک پیام به‌دست آورند. با استفاده از این روش، وانگ و همکارانش توانستند برخوردهایی برای MD۵، در مدت یک ساعت روی یک ابررایانه IBMP۶۹۰ پیدا کنند. در سال ۲۰۰۶ مقاله‌ای توسط بلک[1] منتشر شد که قادر به یافتن برخورد دو بلوک پیام با متوسط یازده دقیقه با محاسبات تابع درهم‌ساز آن با یک رایانه معمولی بود[۱۲]. حدودا در همان زمان، آقای کلیما[2] مقاله‌ی را انتشار داد و روش جدیدی به‌نام "تونل‌زنی" را معرفی کرد که این روش، زمان جستجوی مورد نیاز برای پیدا کردن یک برخورد روی MD۵ را حدودا به ۳۱ ثانیه کاهش داد[۱۳]. در سال ۲۰۰۷ استیون[3] رساله خود را منتشر کرد. او در این رساله به‌طور مفصل چگونگی تولید برخوردها در حدود ۶ ثانیه را برای MD۵ توضیح داده و الگوریتمی را برای تولید مسیرهای جدید تفاضلی آن ارائه داد[۱۴]. تا پایان سال ۲۰۰۸ حمله‌های زیادی بر روی MD۵ انجام گرفت، نهایتا این حمله‌ها منجر به شکست کامل MD۵ شد. بعد از آن، تابع SHA-۱ مورد توجه تحلیل‌گران قرار گرفت. SHA-۱ از لحاظ ساختاری شبیه به MD۵ است اما ویژگی‌هایی دارد که باعث شده این تابع درهم‌ساز امنیت بیشتری در مقایسه با MD۵ داشته باشد. در سال ۱۹۹۵ تابع درهم‌ساز SHA-۱ توسط موسسه NIST منتشر گردید. با این دیدگاه که تابع درهم‌ساز SHA-۱ از نظر ساختاری بسیار شبیه به تابع درهم‌ساز MD۵ بود، لذا حملاتی که به MD۵ موثر بودند، می‌توانستند برای تابع درهم‌ساز SHA-۱ یک تهدید باشند. وانگ و همکارانش یک برخورد را برای کل ۸۰ دور این تابع با کمتر از 2^{69} محاسبه تابع فشرده‌ساز به‌دست آوردند که این زمان، از محاسبات حمله جستجوی کامل (2^{80}) کمتر است[۱۵]. کن‌نیر[4] و رچبرگر[5] روش وانگ و همکارانش را توسعه داده و توانستند روی ۷۰ دور تابع درهم‌ساز SHA-۱ با تعداد 2^{44} محاسبه تابع فشرده‌ساز به یک برخورد برسند. یک مسیر تفاضلی کامل با پیچیدگی از درجه 2^{52} روی ۸۰ دور تابع درهم‌ساز SHA-۱ در رمز اروپایی ۲۰۰۹ ارائه گردید[۱۶]. بعد از این حملات، جامعه رمزنگاری به این باور رسید که تابع درهم‌ساز SHA-۱ از لحاظ تئوری شکسته شده است.

۳-۱- شروع مسابقه SHA-۳

به‌دنبال حملات تاثیرگذاری که روی توابع درهم‌ساز خانواده MD و SHA-۱ ارائه و منجر به شکستن آن‌ها شد و از طرفی چون خانواده SHA-۲ از لحاظ ساختاری شبیه به خانواده MD و SHA-۱ بود لذا حمله‌های موثر به این دو خانواده از توابع درهم‌ساز می‌توانستند تهدیدی جدی برای خانواده SHA-۲ به‌حساب آیند. بدین جهت جامعه جهانی به‌شدت دنبال آن بود تا معیارهایی با سطح امنیتی بالا و سرعت مناسب برای طراحی الگوریتم توابع درهم‌ساز جدید ارائه دهد. برای همین منظور NIST در سال ۲۰۰۷، مسابقه‌ای را به‌نام « مسابقه SHA-۳» آغاز کرد که قرار است تا سال ۲۰۱۲ ادامه داشته باشد. هدف این مسابقه، انتخاب یک تابع درهم‌ساز امن است.

معیارهای NIST برای طراحی الگوریتم تابع درهم‌ساز جدید به‌صورت زیر است:

۱- از نظر ساختاری:

- الگوریتم مورد نظر عمومی بوده و دارای هیچ پارامتر محرمانه و قابل تغییری نباشد،
- الگوریتم مورد نظر دارای قابلیت پیاده‌سازی از نظر سخت‌افزاری و نرم‌افزاری باشد،
- قابلیت ورودی مورد پذیرش در الگوریتم تا طول 2^{64} - 1 بیت را دارا باشد،
- طول خروجی الگوریتم (خلاصه پیام) با اندازه‌های ۲۲۴، ۲۵۶، ۳۸۴ و ۵۱۲ بیت را فراهم نموده و دامنه آن از ۲۲۴ بیت کمتر نباشد.

۲- از نظر امنیتی:

- در مقابل حملات شناخته شده (تحلیل‌های تفاضلی) مقاوم باشد،
- مقاوم در برابر سه مسئله امنیتی تصادم، پیش‌تصویر و پیش‌تصویر دوم باشد،
- دارای رفتاری شبیه به HMAC[6] و دنباله‌های شبه تصادفی (PRNG[7]) باشد،

۳- دارای هزینه‌های مفید و مینیمال باشد،

۴- دارای حداقل حافظه باشد،

۵- الگوریتم انعطاف‌پذیر بوده و ساختاری ساده داشته باشد.

شرایط فوق، حداقل‌های لازم برای توابع درهم‌ساز کاندید شده بوده که در سال ۲۰۰۸ انتشار یافت.

بعد از شروع دور اول مسابقه، NIST در سال ۲۰۰۹، اولین کنفرانس SHA-۳ را برگزار کرد[۱۸و۱۷]. در نهایت NIST، برای دور سوم مسابقه، ۵ نامزد از میان ۱۴ نامزد دور دوم را برای دور نهایی مسابقه انتخاب کرد، اسامی این نامزدها عبارتند از: Blake،

1- Black
2- Klima
3- Stevens
4- Cannière
5- Rechberger

6- Hash MAC
7- Pseudo random generator

Skein، Keccak، Grøstlو JH در نهایت تا سال ۲۰۱۲ یکی از این توابع به‌عنوان تابع درهم‌ساز جهانی معرفی خواهد شد. در ادامه این مقاله ما به معرفی مختصر تابع گروستال پرداخته و سپس به‌عنوان یک کار جدید، حمله مکعبی را بیان و به عدم تاثیر آن بر تابع فوق می‌پردازیم. دلیل انتخاب گروستال به این خاطر است که به نظر ما این تابع در نهایت پذیرفته خواهد شد.

۴- ساختار تابع درهم‌ساز گروستال

این تابع درهم‌ساز به‌عنوان یک نامزد برای $\mathbf{SHA-3}$ معرفی شده است. گروستال [۱۹] یک تابع درهم‌ساز تکرار شونده، با یک تابع فشرده‌ساز است که از دو جایگشت مجزا و ثابت استفاده می‌نماید. در عمل، جایگشت این تابع مشابه یک $S-box$ می‌باشد.

۴-۱- مشخصه تابع گروستال

گروستال مجموعه‌ای از توابع درهم‌ساز با چکیده پیام‌های متفاوت از ۱ تا ۶۴ بایت است. اگر خروجی n بیت مد نظر باشد تابع گروستال به‌صورت $Grostl-n$ نمایش داده می‌شود. گروستال از یک تابع فشرده‌ساز f به‌صورت زیر استفاده می‌کند،

$$f : \{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l$$
$$h_i \leftarrow f(h_{i-1}, m_i) \quad ; i = 1, \ldots, t$$

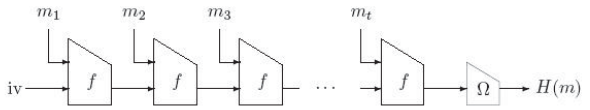
در ابتدا پیام M، طوری لایه‌گذاری می‌شود تا به t قطعه پیام l بیتی $m_1, \ldots, m_t$ تبدیل شود. تابع فشرده‌ساز دو ورودی می‌گیرد که اولی ورودی زنجیر[1] و دومی قطعه‌پیام[2] است. برای آغاز فرایند درهم‌سازی، مقدار اولیه $h_0 = IV$ با یک مقدار اولیه l بیتی، مقدارگذاری می‌شود. که در آن، $n \leq l$ است. بعد از پردازش آخرین بلوک چکیده پیام M به‌صورت زیر محاسبه می‌گردد:

$$H(M) = \Omega(h_t)$$

که در آن Ω تابع انتقال خروجی[3] است. فرض کنید $trunc_n(x)$ عمل‌گری است که از ورودی x فقط n بیت آن را نگه می‌دارد و بقیه آن را دورریز می‌کند. در این صورت تابع انتقال خروجی گروستال عبارت است از:

$$\Omega(x) = trunc_n(P(x) \oplus x)$$

شکل (۲) چگونگی فرایند عمل درهم‌سازی پیام توسط تابع گروستال را نمایش می‌دهد.

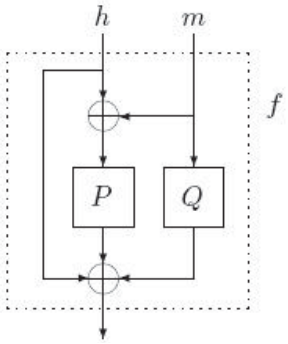


شکل ۲- تابع درهم‌ساز گروستال[۱۹]

تابع فشرده‌ساز بر مبنای دو جایگشت P و Q به‌صورت زیر تعریف می‌شود.

$$f(h,m) = P(h \oplus m) \oplus Q(m) \oplus h$$

شکل (۳) ساختار تابع فشرده‌ساز استفاده شده در تابع درهم‌ساز گروستال را نشان می‌دهد.



شکل ۳- ساختار تابع فشرده‌ساز گروستال[۱۹]

در طراحی جایگشت‌های P و Q از الگوریتم رمز قالبی رایجندال[4] و در استفاده از این جایگشت‌ها از رمز $\mathbf{AES}$ الهام گرفته شده است. در ادامه به یکی از حملات بسیار موثر بر روی توابع درهم‌ساز اشاره نموده و عدم کارایی این حمله بر تابع درهم‌ساز گروستال را بیان و اثبات می‌کنیم.

۵- حملات مکعبی[5]

به موازات توسعه توابع درهم‌ساز، حملات بر این توابع نیز توسعه پیدا نموده که جدیدترین این حمله‌ها، حملات مکعبی است که در سال ۲۰۰۸ توسط آقایان شامیر و دینر انتشار یافت[۲۰]. این حمله را می‌توان از نوع حملات عمومی دانست چرا که می‌تواند به هر سیستم رمزنگاری با دو ورودی مخفی و عمومی اعمال گردد و برای استخراج ورودی مخفی به‌کار رود. به‌علاوه، برای موفقیت حمله، اطلاع از نوع سیستم رمزنگاری لازم نبوده و می‌توان سیستم رمزنگاری را مانند یک جعبه سیاه در نظر گرفت. این حملات به اندازه چندجمله‌ای‌های بیت‌های خروجی وابسته نبوده اما به درجه این چندجمله‌ای‌ها وابسته است. در ادامه بحث، جزئیات این حمله بیان خواهد شد.

1- Chaining input
2- Message block
3- Output transformation

4- Rijndael
5- Cube Attacks

۵-۱- تعاریف و مقدمات

در حملات مکعبی نیاز به تعدادی چندجمله‌ای روی میدان $GF(۲)$ است و هر بیت خروجی تابع درهم‌ساز متناظر با یک چندجمله‌ای می‌باشد. این چندجمله‌ای‌ها، ورودی‌های به‌صورت عمومی و مخفی برای تولید بیت خروجی که متناظر با یک چندجمله‌ای است را می‌پذیرند. بنابراین، این حمله برای یک سیستم با دو ورودی (مثلاً کلید و متن اصلی) ممکن است کارساز باشد.

نکته: چندجمله‌ای $p(x_۱,\dots,x_n)$ و مجموعه $I \subseteq \{۱,\dots,n\}$ که اندیس‌های متغیر p هستند را در نظر گرفته و t_I را جمله‌ای که از حاصل‌ضرب متغیرهایی که اندیس این متغیرها در مجموعه I قرار داشته به‌دست آمده باشد، آنگاه چندجمله‌ای p را می‌توان به‌صورت زیر تجزیه نمود.

$$(x_۱,\dots,x_n) \equiv t_I.P_{S(I)} + q(x_۱,\dots,x_n)$$

که $P_{S(I)}$ یک ابرجمله[1] از I در p است و در آن هیچ‌کدام از متغیرهای t_I وجود ندارند.

تعریف: t_I را ماکسترم[2] گوییم هرگاه درجه $P_{S(I)}$ برابر یک باشد.

تعریف: فرض کنیم I یک مجموعه k عضوی باشد. این k، عضو مکعب بولی k بعدی را تولید می‌کند که گوشه‌های این مکعب متناظر است با همه صفر و یک‌هایی که اعضای این مجموعه می‌توانند بگیرند. این مکعب را می‌توان به‌وسیله یک مجموعه با ۲^k بردار متناظر با گوشه‌های مکعب نشان داد. این مجموعه را با C_I نمایش می‌دهیم. برای یک بردار $v \in C_I$، P_{Iv} را چندجمله‌ای که از P استخراج شده تعریف می‌کنیم به‌طوری‌که متغیرهایی که اندیس آن‌ها در I قرار داشته و با بردار v مقداردهی شده باشند. فرض کنیم $P_I \triangleq \sum_{v \in C_I} P_{Iv}$.

۵-۲- تئوری حمله مکعبی

حمله مکعبی به سیستمی اعمال می‌شود که دارای دو ورودی مخفی و عمومی بوده و مهاجم هیچ دانشی در مورد سیستم رمزنگاری یا تابع درهم‌ساز ندارد اما او به دو چیز دسترسی دارد:

۱. یک شبیه‌ساز[3] که جفت ورودی‌های مخفی و عمومی را گرفته و یک خروجی تولید کند،

۲. یک اوراکل[4] که کلید مخفی داشته و هنگامی که ورودی عمومی به آن داده شود، یک خروجی تولید کند.

قضیه: برای هر چندجمله‌ای P و مجموعه I، هم‌نهشتی $P_I \equiv P_{S(I)}$ در مد ۲ همواره برقرار است.

۵-۳- مشاهدات

- مقدار ابرجمله $P_{S(I)}$ روی بیت‌های مخفی ورودی برابر با P_I است.
- ابرجمله هر ماکسترم t_I خطی است،
- اگر روی بیت‌های ورودی عمومی، ماکسترم‌هایی تولید کنیم و همه بیت‌های عمومی دیگر را برابر صفر قرار دهیم، در صورتی‌که ابرجمله‌های متناظرشان وجود داشته و ثابت نباشند، آنگاه مجموعه چندجمله‌ای‌های خطی روی بیت‌های ورودی برای تشکیل سیستم‌های معادلات خطی را داریم.
- برای یک چندجمله‌ای P از درجه d، درجه هر ماکسترم حداکثر $d-۱$ است.
- محاسبه P روی هر مکعب C_I نیازمند $۲^{|I|}$ مقدار P است.

از آنجا که $|I| \leq d-۱$ برای هر ماکسترم t_I، پس پیچیدگی محاسباتی یک مقدار ابرجمله برابر $O(۲^{d-۱})$ محاسبه P است.

۵-۴- حمله مکعبی به‌طور خلاصه

به‌طور کلی حمله را می‌توان به‌صورت زیر پی‌ریزی نمود.

- پیدا کردن ماکسترم‌ها روی بیت‌های ورودی عمومی با ابرجمله مستقل خطی، روی بیت‌های ورودی مخفی به‌وسیله شبیه‌سازی نمودن تابع هدف با ورودی‌های مخفی و عمومی،
- برای هر ابرجمله روی ورودی‌های مخفی، درخواست از اوراکل برای هر بردار در مکعب ماکسترم، و سپس ترکیب ابرجمله‌ها و مقادیرشان برای تشکیل یک سیستم از معادلات خطی،
- حل سیستم معادلات خطی بر حسب ورودی‌های مخفی.

۵-۵- پیش محاسبه[5]

یک حمله مکعبی می‌تواند به دو فاز مجزا تقسیم شود. مرحله پیش‌محاسبه و مرحله حمله آنی[6]. پیش محاسبه، قسمتی از حمله است که مستقل از اوراکل می‌تواند انجام شود. حمله آنی، فازی است که نتایج حاصل از پیش‌محاسبه برای سؤال پرسیدن از اوراکل و استخراج کلید مخفی استفاده می‌شوند. در فاز پیش‌محاسبه، هدف، پیدا کردن ماکسترم‌هایی است که متناظر با ابرجمله‌ها روی بیت‌های مخفی و استخراج ساختار جبری صحیح این ابرجمله‌ها است. کافی است که فقط ابرجمله‌های مستقل خطی تولید شوند. آن‌ها می‌توانند در همه مراحل حمله‌های آنی روی این سیستم ذخیره شده و مورد استفاده قرار گیرند.

1- Superpoly
2- Maxterm
3- Simulator
4- Oracle
5- Precomputation
6- Online

۵-۶- استخراج ابرجمله

یک تناظر یک‌به‌یک بین ماکسترم و ابرجمله وجود دارد. از این‌که هیچ دانشی از ساختار جبری سیستم رمزنگاری نداریم، لذا باید ساختار جبری یک ابرجمله را از ماکسترمش استخراج کنیم. برای یک ابرجمله داده شده، می‌دانیم آن ترکیبی از جمله‌هایی که فقط یک متغیر x_j دارند برای هر بیت مخفی با اندیس j و یک جمله ثابت است. (وجود جمله ثابت تضمین شده نیست). بنابراین ساختار جبری هر ابرجمله می‌تواند به‌وسیله پیدا کردن ضرایب برای همه x_j ها و مقدار جمله ثابت به‌کار گرفته شود. این مقادیر می‌توانند به‌صورت زیر تعیین شوند.

جمله ثابت: مقدار P_I را محاسبه می‌کنیم بدین صورت که همه ورودی‌ها را برابر صفر قرار می‌دهیم به‌جز آن متغیرهایی که اندیس‌شان در I قرار دارد.

ضرایب x_j: محاسبه مقدار P_I با جایگزین کردن همه ورودی‌ها با صفر به‌جز آن ورودی‌هایی که اندیس‌شان در I قرار دارد و x_j را برای این ورودی‌ها، مقدار یک می‌دهیم. بنابراین ضریب x_j به این مقدار و ثابت ابرجمله بستگی دارد. بدین‌صورت که ابرجمله شامل x_j است اگر و فقط اگر نتیجه حاصل، مخالف نتیجه حاصل از جمله ثابت باشد.

۵-۷- پیدا کردن ماکسترم‌ها

الف) پیدا کردن ماکسترم برای چندجمله‌ای‌های تصادفی

تعریف: یک چندجمله‌ای مانند p از درجه d با $n+m$ متغیر یک چندجمله‌ای تصادفی می‌باشد، هرگاه هر جمله با درجه حداکثر d به‌طور مستقل و با احتمال یکنواخت انتخاب گردد.

تعریف: یک چندجمله‌ای تصادفی d با $n+m$ متغیر، چندجمله‌ای مانند p می‌باشد به‌طوری که هر جمله ممکن از درجه d شامل یک متغیر مخفی و $d-1$ متغیر عمومی باشند، که به‌طور مستقل با احتمال یکنواخت انتخاب شده و همه جمله‌های دیگر بتوانند به‌صورت دلخواه با هر احتمالی انتخاب شوند.

در یک چندجمله‌ای تصادفی d، هر جمله t_I که از ضرب $d-1$ متغیر عمومی تولید شده است با احتمال بالایی یک ماکسترم می‌باشد که ابرجمله متناظرش یک چندجمله‌ای از درجه حداکثر ۱ می‌باشد.

ب) پیدا کردن ماکسترم برای چندجمله‌ای‌های غیرتصادفی

به‌منظور پیدا کردن ماکسترم‌ها، بهترین شرایط جستجو برای آن‌هایی است که به درجه چندجمله‌ای یعنی d نزدیک باشند. یک روش، گام تصادفی و جستجوی کامل است. بعد از آن‌که به اندازه کافی ابرجمله‌های مستقل خطی پیدا شدند، حمله آنی می‌تواند آغاز شود.

ساختار حمله آنی عبارت است از:

۱. برای هر چندجمله‌ای خطی $P_{S(I)}$، مقدار این چندجمله‌ای را با استفاده از قضیه فوق و با درخواست از اوراکل یافته،

۲. ترکیب نمودن مقادیر فوق با ابرجمله‌ها به‌منظور تشکیل یک سیستم معادلات خطی بر حسب بیت‌های ورودی که در اوراکل پنهان شده است،

۳. در نهایت حل سیستم معادلات فوق.

۵-۸- پیچیدگی حمله

همان‌طور که قبلا ذکر شد پیچیدگی محاسباتی یک حمله مکعبی محدود به درجه چندجمله‌ای بیت‌های خروجی سیستم تحت حمله می‌باشد. از آنجا که پیچیدگی یک حمله به فاز لحظه‌ای حمله بستگی دارد، در ابتدا فاز لحظه‌ای را در نظر می‌گیریم.

چندجمله‌ای P با درجه d و n بیت ورودی مخفی و m بیت ورودی عمومی را در نظر می‌گیریم. پیچیدگی محاسبه مقدار n ابرجمله حداکثر برابر n^{d-1} با درخواست از اوراکل است. پیچیدگی حل سیستم معادلات خطی $O(n^3)$ است. بنابراین کل پیچیدگی حمله آنی برابر $O(n^{d-1})+O(n^3)$ می‌باشد.

همچنین برای فاز پیش‌محاسبه، از تعدادی ارزیابی‌های مکعب که هر کدام نیازمند $O(2^{d-1})$ استفاده از شبیه‌ساز سیستم است، تشکیل شده است. تعداد دقیق محاسبه‌های مکعب وابسته به شیوه استفاده شده در پیدا کردن ماکسترم‌ها و همچنین تعداد ماکسترم‌های حدس‌زده‌شده‌ای که مورد قبول واقع نمی‌گردند می‌باشد، که به ساختار سیستم بر می‌گردد.

۱- حمله مکعبی روی تابع فشرده‌ساز گروستال

هدف این بخش، اعمال حمله مکعبی روی تابع فشرده‌ساز گروستال با تعداد دورهای کاهش‌یافته می‌باشد.

برای نشان دادن نحوه عمل‌کرد این حمله مثال زیر را در نظر بگیرید: $p(v_0,v_1,v_2,x_0,x_1,x_2)$ که متغیرهای (v_0,v_1,v_2) ورودی‌های عمومی هستند و در دسترس مهاجم می‌باشند ولی متغیرهای (x_0,x_1,x_2) ورودی‌های مخفی هستند که مهاجم هیچ‌گونه اطلاعاتی در مورد این ورودی‌ها ندارد و هدفش به‌دست آوردن این اطلاعات می‌باشد، چندجمله‌ای فوق به‌صورت زیر تعریف شده‌است (مهاجم هیچ‌گونه اطلاعاتی در مورد ساختار این چندجمله‌ای ندارد، در اینجا فقط برای روشن شدن مطلب و نحوه حمله، ساختار چندجمله‌ای آورده شده است):

$$p(v_0,v_1,v_2,x_0,x_1,x_2)=v_0v_1x_0+v_0v_1x_1+v_2x_0x_2 \\ +v_1x_2+v_0x_0+v_0v_1+x_0x_2+v_1+x_2+1$$

در ابتدای امر، مهاجم باید ماکسترم‌های مورد نظر را پیدا کند، از توضیحات قبل می‌دانیم که ماکسترم مورد نظر باید زیرمجموعه‌ای از جملات عمومی باشد، لذا باید زیرمجموعه‌ای را انتخاب کنیم که ابرجمله آن خطی و غیر ثابت باشد. از آنجا که هدف، نشان دادن چگونگی حمله می‌باشد فرض می‌کنیم ماکسترم $t_{I_0} = v_0 v_1$ که $I_0 = \{0,1\}$ را مهاجم انتخاب نماید. بنابراین،

$$P_{I_0} = p(0,0,v_2,x_0,x_1,x_2) + p(0,1,v_2,x_0,x_1,x_2) + p(1,0,v_2,x_0,x_1,x_2) + p(1,1,v_2,x_0,x_1,x_2)$$

همان‌طور که توضیح داده شد، المان‌هایی از ورودی‌های عمومی که در ماکسترم نیستند را می‌توان ثابت فرض کرد، مثلا برابر با صفر در نظر گرفت، لذا داریم:

$$P_{I_0} = p(0,0,0,x_0,x_1,x_2) + p(0,1,0,x_0,x_1,x_2) + p(1,0,0,x_0,x_1,x_2) + p(1,1,0,x_0,x_1,x_2)$$

مهاجم به چندجمله‌ای p دسترسی ندارد، اما با استفاده از شبیه‌ساز می‌تواند ثابت نبودن و خطی بودن P_{I_0} را بررسی کند.

برای تست ثابت نبودن، مهاجم از شبیه‌ساز می‌خواهد که متغیرهای مخفی را برابر با $(0,0,0)$ و $(1,0,1)$ قرار بدهد و مقدار P_{I_0} را به مهاجم برگرداند، شبیه‌ساز مقادیر زیر را به مهاجم بر می‌گرداند:

$$P_{I_0}(0,0,0) = 1 \quad P_{I_0}(1,0,1) = 0$$

با توجه به مقادیر برگردانده شده، مهاجم به این نتیجه می‌رسد که ابرجمله حاصل از این ماکسترم ثابت نیست لذا به مرحله بعد می‌رود. برای اثبات خطی بودن، یک‌بار دیگر مهاجم از شبیه‌ساز درخواست می‌کند که برای مقادیر مختلف ورودی‌های مخفی تست خطی بودن را انجام بدهد(در این مرحله باید تعداد آزمایش‌ها برای مقادیر مختلف ورودی‌های مخفی و تعداد نتیجه خطی بودن به‌قدری زیاد باشد که مهاجم تقریبا به قطعیت خطی بودن ابرجمله برسد):

برای مثال شبیه‌ساز تست زیر را انجام می‌دهد:

$$P_{I_0}(0) + P_{I_0}(x) + P_{I_0}(y) = P_{I_0}(x \oplus y)$$
$$P_{I_0}(0,0,0) + P_{I_0}(1,0,1) + P_{I_0}(0,1,1) = P_{I_0}(1,1,0)$$
$$1 + 0 + 0 = 1$$

اگر مهاجم برای مقادیر دیگر نیز امتحان کند به همین نتیجه می‌رسد، لذا با قطعیت می‌تواند بگوید این ابرجمله خطی است.

روند حمله ما نیز به تابع درهم‌ساز گروستال مشابه روش بالا است. از آنجا که این حمله برای سیستم‌های رمزنگاری با دو نوع ورودی مخفی و عمومی قابل اعمال می‌باشد، لذا ما فرض می‌کنیم که مهاجم ۱۶ بیت ابتدایی از بلوک اول را نمی‌داند $(x_0, x_1, \ldots, x_{15})$ و بقیه بیت‌های همان بلوک را می‌داند و هدف او پیدا کردن ۱۶ بیت مخفی می‌باشد. در واقع این حمله، حمله پیش‌تصویر جزیی می‌باشد. در این حمله فرض شده که تعداد دورهای تابع فشرده‌ساز ۳ و پیام ورودی یک قطعه ۵۱۲ بیتی باشد. ما حمله مکعبی را با استفاده از نرم‌افزار متلب شبیه‌سازی نموده و آن را برای تابع گروستال با مفروضات فوق اجرا نمودیم. برای پیدا کردن ماکسترم‌ها، به‌صورت تصادفی اندیس‌های مکعب را انتخاب کردیم و برای هر مکعب شرط خطی بودن و ثابت نبودن ابرجمله متناظر با این مکعب را اعمال کرده (مانند روند فوق) و در نهایت، نتایج حاصل در جدول (۱) ثبت شده است. قابل ذکر است که ما جملات زیادی را برای ماکسترم بودن امتحان کردیم ولی برای تمام این جملات به‌طور قطعی نتوانستیم نتیجه‌گیری کنیم که ابرجمله حاصل از این ماکسترم‌ها خطی می‌باشد یا غیرخطی. تقریبا برای تمام انتخاب‌ها می‌توان نتیجه گرفت که جمله حاصل تقریباً با احتمال یکنواخت خطی است و با همین احتمال غیرخطی. که این احتمال هیچ اطلاعاتی به مهاجم نمی‌دهد. حتی ما تابع درهم‌ساز گروستال را ساده‌تر کردیم طوری که فقط حمله را روی سه دور جایگشت Q اعمال کردیم که در این مورد نیز حمله بر روی آن بی‌تاثیر بود. نتیجه این‌که جایگشت‌های استفاده شده در تابع گروستال تقریبا ایده‌آل هستند.

قابل ذکر است که در [۲۱] حمله‌های موفقیت‌آمیزی روی دو تابع درهم‌ساز کیکا و اس‌سنک که دورهای آن‌ها کاهش داده شده صورت گرفته است. به نظر می‌رسد، مهمترین دلیل موفقیت حمله مکعبی روی این دو تابع درهم‌ساز، استفاده نکردن این دو تابع از S-box است در حالی که در تابع گروستال از P-boxهایی استفاده شده که در عمل مانند یک S-box خوب عمل می‌نمایند.

در جدول (۱)، مهاجم ابرجمله‌های متناظر با هر ماکسترم را با کمک اوراکل به‌دست آورده است، اما همان‌طور که ذکر شد در هیچ‌یک از این موارد نمی‌تواند به‌طور قطعی، خطی بودن این ابرجمله‌ها را نتیجه بگیرد و حل این معادلات هیچ‌گونه اطلاعاتی در مورد مقادیر مخفی نمی‌دهد.

۷- نتیجه‌گیری

در این مقاله، ما مفاهیم امنیت و سندیت یک پیام با روی‌کرد پدافند غیرعامل را بیان نموده و برای هر کدام ابزار مورد نیاز را به‌طور مفصل شرح دادیم. یکی از مقوله‌های مهم در پروتکل‌های ارتباطی، توابع

درهم‌ساز است. خصوصیات کلی این توابع را مورد بررسی قرار داده و در ادامه به یکی از مهم‌ترین توابع درهم‌ساز جدید به نام گروستال از نظر ساختاری پرداختیم. در پایان، حمله مکعبی که یک حمله جدید و مؤثر است را توضیح دادیم و این حمله را به فشرده‌ساز تابع گروستال اعمال نمودیم. نتیجه آن، عدم کارایی این حمله به تابع گروستال بوده است.

جدول ۱- ماکسترم‌ها و مقدار متناظر آن‌ها برای سه دور از تابع فشرده‌ساز گروستال

ابرجمله و مقدار متناظر آن	اندیس‌های مکعب	بیت خروجی
$1+x_1+x_2+x_3+x_6+x_7+x_{13}=0$	$\{30,60,90,120\}$	411
$x_5+x_7+x_{10}+x_{11}+x_{13}+x_{16}=0$	$\{102,168,200,400\}$	310
$1+x_1+x_7+x_8+x_{11}+x_{12}+x_{13}+x_{15}=1$	$\{32,40,48,56,112\}$	300
$1+x_2+x_4+x_7+x_{11}+x_{15}+x_{16}=0$	$\{99,140,226,333,433\}$	199
$1+x_1+x_2+x_4+x_5+x_6+x_{10}+x_{11}+x_{14}+x_{16}=1$	$\{89,109,189,268\}$	199
$1+x_1+x_2+x_4+x_9+x_{10}+x_{11}=0$	$\{17,117,217,317,417\}$	151
$1+x_1+x_2+x_3+x_4+x_6+x_7+x_8+x_9+x_{10}+x_{16}=0$	$\{166,199,250,350\}$	45
$x_1+x_2+x_3+x_4+x_8+x_{11}+x_{12}+x_{13}+x_{16}=0$	$\{40,120,240,369,427\}$	402
$x_1+x_2+x_4+x_8+x_9+x_{11}+x_{12}+x_{13}+x_{14}+x_{15}=0$	$\{77,87,97,103,114\}$	32
$1+x_2+x_5+x_6+x_9+x_{11}+x_{13}+x_{15}+x_{16}=0$	$\{70,102,134,164,221\}$	78
$1+x_1+x_{12}+x_{13}=0$	$\{91,181,233,333,401\}$	89
$x_1+x_3+x_4+x_8+x_{12}+x_{14}+x_{15}+x_{16}=0$	$\{77,118,210,315,401\}$	277
$1+x_3+x_5+x_6+x_7+x_8+x_{12}+x_{14}=1$	$\{411,419,431,495\}$	25
$x_1+x_3+x_6+x_7+x_{10}+x_{12}+x_{13}=1$	$\{46,88,95,104,115\}$	27
$1+x_1+x_3+x_4+x_9+x_{11}+x_{13}+x_{16}=1$	$\{201,207,216,309,485,505\}$	295
$x_1+x_2+x_8+x_{10}+x_{11}+x_{13}+x_{14}+x_{15}=1$	$\{37,58,171,262,373,502\}$	366

مراجع

1. T. Andrews, “Computer Networks (FOURTH EDITION)”.
2. R. Rivest. The MD5 message-digest algorithm. Technical report, IETF, (**1992**).
3. National Institute of Standard and Technology "Descriptions of SHA-256, SHA-384, and SHA-512"،PREPRINT, (**2000**), pp.1-48.
4. I. Damgard, “A Design Principle for Hash Functions”, In Advances in Cryptology, Crypto89, volume 435 of LNCS, PAGES 56-71,Springer-Verlag, (**1989**).
5. R. Merkle, “One Way Functions and DES”, In Advances in Cryptology, Crypto89, volume 435 of LNCS, Springer-Verlag, (**1990**).
6. B. Preneel, R. Govaerts, and J. Vandewalle, "Hash Function Based on Block Ciphers:A synthetic Approach", Advances in Cryptology-CRYPTO'93 proceedings. Springer Verlag،(**1994**)، pp.368-378.
7. W. Stallings, Network and Internetwork Security .Principles and Practice "،Prentice Hall ، Inc. (**1995**).
8. Federal Information Processing Standards (FIPS) Publication 180-1, Secure Hash Standard (SHS), U.S.DOC/NIST, April 17, (**1995**).
9. Draft Federal Information Processing Standards (FIPS) Publication 180-1, Secure Hash Standard (SHS), U.S.DOC/NIST, April 17, (**1995**).
10. B. Schniere, "Applied Cryptography Second Edition Protocols, Algorithms, and Source Code in c ", john wiley & sons, inc. (**1996**).
11. X. Wang and H. Yu, How to break MD5 and other hash functions. Lecture Notes in Computer Science, 3494, (**2005**).
12. J. Black and M. Cochran, A study of the MD5 attacks: Insights and improvements In Fast Software Encryption, pages 262-277. SpringerVerlag, (**2006**).
13. V. Klima, Tunnels in hash functions: MD5 collisions within a minute. Cryptology ePrint Archive, Report 2006/105, (**2006**).
14. M. Stevens, On collisions for MD5. Master's thesis, Eindhoven University of Technology, (**2007**).
15. X. Wang, Y. L. Yin and H. Yu, Finding collisions in the full SHA-1. Lecture Notes in Computer Science, 3621, (**2005**).

16. C. De. Canniere and C. Rechberger. Finding SHA-1 characteristics, Lecture Notes in Computer Science, 4284, (**2006**).
17. NIST SHA-3 competition first round candidates „ (**2009**). .http://csrc.nist.gov/groups /St /hash /sha- 3/Round1/ submissions_rnd1.html,**2009**
18. NIST SHA-3 competition round one webpage. http://csrc.nist.gov/ groups/ST/hash/sha-3/Round1/index. html, (**2009**).
19. P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. Grøstl - a SHA-3 candidate.Submission to NIST, (**2008**).
20. I. Dinur and A. Shamir, Cube attacks on tweakable black box polynomials. Cryptology ePrint Archive, Report 2008/385, **2008**. http://eprint.iacr.org/.
21. J. Lathrop," Cube Attacks on Cryptographic Hash Functions", (**2009**).

The Role of Cryptographic Scrambler Functions in Security with a Passive Defense Approach

M. A. Taheri[1]

Z. A. Noroozi[2]

Abstract

Information and communications security is of utmost importance in both military and security fields. Establishing an appropriate and improved cover to safeguard security and authenticity in communications and information matters through cryptography and communication protocols is possible. The mechanisms which contribute to the security of the transmission and reception of a message are related to passive defense and will contribute to the reduction of information vulnerability. On the other hand, the communication protocols which review the authenticity of a piece of communication or information, cause the reduction of access of unauthorized personnel to confidential information and, moreover, will reduce the vulnerability of a communication network through processes such as lack of saving huge files, lack of access to parts of a message, creating the message summary with a very short length for each long secret and non-secret message and so on. The above process is a very precise concept of passive defense, because it will dramatically reduce the communication and information vulnerability in communication protocols. The security of a message through cryptography is obtainable. In many cases, the aim of the sender and the receiver of a message is the authenticity and integrity of the message. This is made possibly through the use of communication protocols such as digital signatures, scrambler functions, message credit codes and so on.

The scrambler functions play a fundamental role in the communication protocols. For instance, failing to use a scrambler function in a digital signature will cause the dysfunction of the signature. In this essay, the exact concept of security and the applications of scrambler functions will be expressed and then the course of the development of these functions will be mentioned, as well.

One of the new functions, called Grostal (one of the five scrambler functions which has reached the third stage of the NIST contest and the one that we think will eventually be selected by this institute as a global scrambler function) will be reviewed in terms of structure and as a new task, the security of this function against cubic attacks will also be analyzed.

Key Words: *Mbler Functions, Message Credit Codes, Cryptography, Cubic Attacks*

1- MS. Candidate of Telecommunications, Imam Hossein Comprehensive University (Email: Taheri.nodh@yahoo.com)

2- Lecturer and Academic Member of Imam Hossein Comprehensive University, Mathematics-Cryptography Department (Email: Znorozi@ihu.ac.ir)