به نام خدا

دانشکده‌ی علوم ریاضی

مقدمه‌ای بررمزنگاری اردیبهشت ۱۳۹۳

جلسه‌ی ۱۸: توابع چکیده‌ساز

مدرس: دکتر شهرام خزائی *نگارنده: مهری رضایی علی‌آبادی*

۱ توابع چکیده‌ساز

تعریف ۱ (تابع چکیده‌ساز) *گوییم $H : \{0,1\}^* \to \{0,1\}^n$ یک تابع چکیده‌ساز[1] است هرگاه، هر پیام ورودی با طول دلخواه را به یک خروجی با طول ثابت (چکیده‌ی پیام[2]) به طور کارایی تصویر کند.*

توابع چکیده‌ساز کاربردهای گسترده‌ای در امضای دیجیتال، پروتکل‌های احراز هویت، حفظ یکپارچگی پیام وتولید اعداد شبه تصادفی دارند. استفاده ازتوابع چکیده‌ساز درالگوریتم های امضای دیحیتال به این صورت است که ابتدا چکیده پیامی که باید امضا شود محاسبه می‌گردد وسپس این چکیده امضا می‌شود. جهت حفظ یکپارچگی پیام‌هایی که بروی کانال های ناامن فرستاده می‌شوند، عموما از توابع چکیده‌ساز استفاده می‌شود، بدین منظورچکیده‌ی پیام باید به "نوعی" تصویری یکتا از پیام ورودی باشد و همچنین در این حالت به سبب کوتاه‌تر بودن طول چکیده نسبت به طول ورودی، حفظ یکپارچگی پیام ساده‌تر خواهد بود. ازکاربردهای دیگر توابع چکیده‌ساز، ذخیره سازی کلمات عبور در سیستم عامل و نرم افزارها است به این صورت که به جای ذخیره کلمه عبور، چکیده‌ی آن دخیره می‌شود، این کار مستلزم آن است که تابع چکیده‌ساز به نحوی باشد که ازروی خروجی آن نتوان ورودی را محاسبه نمود. به طور کلی ازیک تابع چکیده‌ساز ایده‌آل انتظار می‌رود ویژگی‌های زیر را داشته باشد.

- **برخوردتابی**[3]: پیدا کردن دوپیام متمایز m و m' که چکیده‌ی یکسانی داشته باشند سخت باشد؛ یعنی از نظر محاسباتی نتوان پیام‌های $m \neq m'$ را یافت که برای آنها $H(m) = H(m')$ باشد.
- **پیش‌تصویرتابی**[4]: به ازای y تصادفی داده شده در برد تابع H، پیداکردن پیام m که برای آن داشته باشیم $H(m) = y$، سخت باشد.
- **پیش‌تصویر دوم‌تابی**[5]: به ازای هر m داده شده، پیدا کردن پیام $m' \neq m$ که $H(m) = H(m')$، سخت باشد.

[1] Hash function
[2] message digest
[3] collision-resistant
[4] preimage-resistant
[5] second-preimage-resistant

۲ امنیت توابع درهم ساز

حمله جستجوی کامل به تمامی توابع چکیده ساز قابل اعمال است. امنیت تابع چکیده‌ساز در مقابل حمله جستجوی کامل متناسب با طول خروجی آن است بنابراین طول خروجی باید به اندازه کافی بزرگ باشد تا تابع چکیده ساز در مقابل حملات زیر امن باشد. فرض کنیم H تابع چکیده‌ساز ایده‌آل با n بیت خروجی باشد.

- **حمله برخورد:** مهاجم سعی میکند دوپیام $m \neq m'$ رابه گونه ای بیابد که برای آن‌ها $H(m) = H(m')$ باشد. با توجه به قضیه تناقض روزتولد، برای آنکه مهاجم بتواند با احتمال بیشتر از ۱/۲ چنین پیام‌هایی را بیابد نیاز به محاسبه $۲^{n/۲}$ پیام ورودی دارد.
- **حمله پیش‌تصویر:** برای یک چکیده‌ی n بیتی y وقتی مهاجم پیام دلخواه m را امتحان می‌کند با احتمال ۲^{-n} وی می‌تواند انتظار داشته باشد که $H(m) = y$ باشد. بنابراین وی نیاز به محاسبه‌ی ۲^n پیام ورودی دارد تا با احتمال خوبی یکی از آنها پیش‌تصویری برای y باشد.
- **حمله پیش‌تصویر دوم:** در این حمله مهاجم سعی می‌کند برای پیام m ای که در اختیار دارد، پیام $m \neq m'$ را به نحوی پیدا کند که $H(m) = H(m')$ باشد، مهاجم برای یافتن چنین پیامی نیاز به محاسبه ۲^n مقدار ورودی دارد.

با توجه به اینکه طول خروجی تابع چکیده‌ساز ثابت، ولی طول ورودی آن دلخواه است بدیهی است که برای تابع چکیده‌ساز، همواره می‌توان برخورد پیدا کرد. امنیت تابع چکیده به این معناست که پیداکردن برخورد ازنظر محاسباتی ممکن نباشد. گوییم یک تابع چکیده‌ساز شکسته شده است اگر الگوریتمی با پیچیدگی کم‌تر از الگوریتم جستجوی کامل بتواند حداقل یکی از ویژگی‌های تابع ایده‌آل را نقض کند.

۳ طراحی توابع چکیده‌ساز

طراحی توابع چکیده‌ساز در ساختار مرکل-دمگارد و اسفنجی از مهم‌ترین روش‌های طراحی توابع چکیده‌ساز می‌باشند. در ادامه به چگونگی طراحی توابع چکیده‌ساز در این ساختارها می‌پردازیم.

۴ ساختار مرکل-دمگارد

نخستین بارمرکل و دمگارد استفاده از توابع فشرده‌ساز[۶] در طراحی توابع چکیده‌ساز را مطرح نمودند. تابع فشرده‌ساز در واقع نوعی تابع چکیده‌ساز است که برای پیام‌های با طول ثابت قابل استفاده است. ایده طراحی تابع چکیده‌ساز در ساختار مرکل-دمگارد این است که با داشتن تابع فشرده‌ساز برخوردتاب $f : \{۰,۱\}^n \times \{۰,۱\}^t \to \{۰,۱\}^n$ می‌توان تابع چکیده‌ساز برخوردتاب $H : \{۰,۱\}^* \to \{۰,۱\}^n$ را با تکرار تابع فشرده‌ساز به صورت بازگشتی طراحی نمود.
بدین منظور ابتدا اگر طول دنباله ورودی به تابع چکیده‌ساز، مضرب صحیحی از طول قالب ورودی تابع فشرده‌ساز نباشد، به انتهای پیام ورودی، یک بیت یک و به تعداد موردنیاز بیت صفر اضافه می‌شود. و در صورتی که طول دنباله ورودی به تابع چکیده‌ساز، مضرب صحیحی از طول قالب ورودی تابع فشرده‌ساز باشد، یک بیت ۱ و $t - ۱$

[۶] compression function

بیت صفر به انتهای پیام ورودی اضافه می‌شود. افزودن بیت های اضافی، به این شیوه را پدینگ گوییم. ساختار تابع چکیده‌ساز در ساختار مرکل-دمگارد را در شکل ۱۷-۱ مشاهده می‌کنیم.

شکل ۱۷-۱: ساختار مرکل-دمگارد

در این ساختار h_i متغیر زنجیره‌ای[۷] نامیده می‌شود و در مرحله اول برابر با یک مقدار اولیه[۸] ثابت IV (مثلاً $IV = 0^n$) است. پیام ورودی m، بعد از اعمال پدینگ روی آن به b قالب t بیتی $m_1, m_2, .., m_b$ تقسیم می‌شود سپس مقدار چکیده برای پیام m به صورت زیر محاسبه می‌شود که $[\ell]_t$ نمایش دودویی طول پیام m با یک رشته t بیتی است.

- $h_0 = IV$
- $h_{i+1} = f(h_i, m_{i+1})$ for $0 \leq i < b$
- $H(m) = f(h_b, [\ell]_t)$

قضیه ۱ *اگر تابع فشرده‌ساز f برخوردتاب باشد آنگاه تابع چکیده‌ساز H نیز برخوردتاب است.*

برهان. فرض کنید تابع چکیده‌ساز H برخوردتاب نباشد بنابراین پیام‌های متمایز m و m' وجود دارند که $H(m) = H(m')$ است. نشان می‌دهیم وجود برخورد برای تابع چکیده‌ساز منجر به یافتن برخوردی برای تابع فشرده‌ساز در جایی از ساختار تکراری می‌شود. فرض کنید قالب‌های پیام‌های پد شده $m_1, \cdots, m_b$ و $m'_1, \cdots, m'_{b'}$ باشند. همچنین طول‌های پیام‌ها را با ℓ و ℓ' و متغیرهای زنجیره‌ای متناظر را $h_0, h_1, \cdots, h_b$ و $h'_0 = h_0, h'_1, \cdots, h'_{b'}$ در نظر بگیرید. بدین منظور دوحالت زیر را در نظر می‌گیریم:

- **طول پیام‌های m و m' برابر است:**

  در این‌صورت $\ell = \ell'$ و $b = b'$ و بنابراین

[۷] chaining value
[۸] Initial Value

$H(m) = f(h_b, [\ell]_t)$
$H(m') = f(h'_b, [\ell']_t)$

اگر $h'_b \neq h_b$ باشد آنگاه برخوردی برای تابع f بدست می‌آید. درغیر اینصورت یک قالب به عقب بر می‌گردیم. حال اگر $(h'_{b-1}, m'_b) \neq (h_{b-1}, m_b)$ باشد آنگاه برخوردی یافته ایم. در غیر این صورت این روند را تا یافتن برخوردی برای f ادامه می‌دهیم. با توجه به اینکه دو پیام متمایزند پس حداقل در یک قالب با هم متفاوت خواهند بود بنابراین در روند بازگشتی حتما برخوردی برای f خواهیم یافت.

- **پیام‌های m و m' طول یکسانی ندارند:**

در این صورت $\ell \neq \ell'$ و بنابراین

$H(m) = f(h_b, [\ell]_t)$
$H(m') = f(h'_{b'}, [\ell']_t)$
$H(m) = H(m') \Rightarrow f(h'_{b'}, [\ell']_t) = f(h_b, [\ell]_t)$

با توجه به اینکه $\ell \neq \ell'$ است بنابراین $f(h'_{b'}, [\ell']_t) = f(h_b, [\ell]_t)$ برخوردی برای تابع f می‌باشد.

■

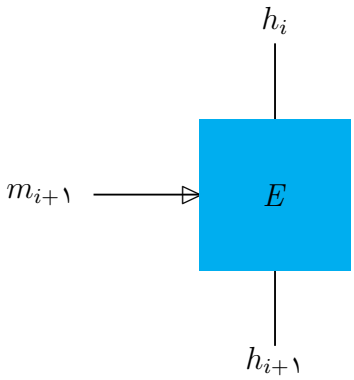
نکته ۱ *عکس گزاره بالا لزومابرقرار نیست و به طور کلی نمی‌توان ضعف‌های تابع چکیده‌ساز را به تابع فشرده‌ساز تعمیم داد.*

نکته ۲ *وارد کردن طول پیام در ساختار مرکل- دمگارد ضروری است. در غیر اینصورت می‌توان توابع فشرده‌ساز برخوردتابی یافت که ساختار مرکل- دمگارد متناظر (بدون قالب طول پیام) برخوردتاب نباشد (راهنمایی: فرض کنید تابع فشرده‌ساز نقطه ثابتی مانند $IV = f(IV, m)$ داشته باشد.)*

نکته ۳ *قضیه مرکل- دمگارد نشان می دهد اگر تابع فشرده‌ساز ایده‌آل باشد به طوری که بهترین حمله برخورد برای آن دارای پیچیدگی $2^{n/2}$ باشد، بهترین حمله برخورد علیه تابع چکیده‌ساز نیز دارای همان پیچیدگی است. با این وجود ساختار مرکل- دمگارد نسبت به پیش‌تصویرتابی ایده‌ال عمل نمی‌کند؛ زیرا حملاتی با پیچیدگی بهتری از 2^n علیه آن وجود دارد.*

۱.۴ طراحی توابع فشرده‌ساز

در بخش قبل دیدیم در صورت وجود یک تابع فشرده‌ساز برخوردتاب، می‌توان تابع چکیده‌ساز برخوردتاب طراحی کرد. حال در این بخش به چگونگی طراحی تابع فشرده‌ساز مقاوم در برابر برخورد می‌پردازیم. یکی از روش های رایج در طراحی توابع فشرده‌ساز استفاده از رمزهای قالبی است. فرض کنیم $E : \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ یک رمزقالبی باشد.
ایده اولیه طراحی تابع فشرده‌ساز f به این صورت است که در رمزقالبی E پیام ورودی را به عنوان کلید اصلی و مقدار h_i را به عنوان متن اصلی ورودی در نظر بگیریم. ساختار تابع f را در شکل ۱۷-۲ مشاهده می‌کنیم.



شکل ۱۷-۲ .

تابع f مطابق رابطه زیر می‌باشد.

$$h_{i+1} = f(h_i, m_{i+1}) = E_{m_{i+1}}(h_i)$$

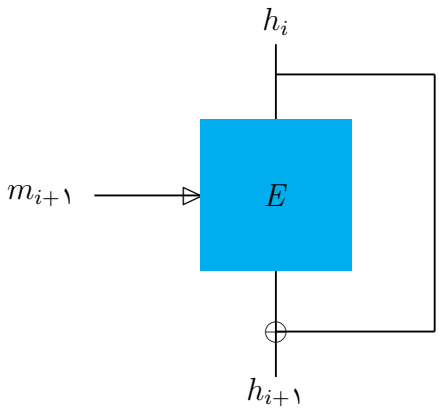
حال سوالی که مطرح می‌شود این است که آیا تابع فشرده‌ساز با ساختار معرفی شده برخوردتاب است؟ پاسخ به این پرسش منفی است. مهاجم می‌تواند به راحتی برای هر خروجی دلخواه h_{i+1} از تابع فشرده‌ساز به تعداد دلخواه پیش‌تصویر بسازد. برای این کار کلید دلخواه m_{i+1} را انتخاب می‌کند و مقدار ${E^{-1}}_{m_{i+1}}(h_{i+1})$ را محاسبه می‌کند. به وضوح $\left(m_{i+1}, {E^{-1}}_{m_{i+1}}(h_{i+1})\right)$ پیش‌تصویری برای h_{i+1} است.
برای ساختن برخورد به عنوان مثال مهاجم خروجی $h_{i+1} = 0^n$ و پیام‌های متمایز $m_{i+1} = 0^n$ و $m'_{i+1} = 1^n$ را درنظر می‌گیرد. حال با محاسبه ورودی‌های h_i و h'_i به صورت زیر برخوردی برای تابع f می‌یابد.

$$\begin{array}{l} h_i = E^{-1}_{m_{i+1}}(h_{i+1}) = {E^{-1}}_{0^n}(0^n) \\ h'_i = E^{-1}_{m'_{i+1}}(h_{i+1}) = {E^{-1}}_{1^n}(0^n) \\ f(h_i, m_{i+1}) = E_{m_{i+1}}(h_{i+1}) = 0^n \\ f(h'_i, m'_{i+1}) = E_{m'_{i+1}}(h'_{i+1}) = 0^n \end{array} \quad \Longrightarrow \quad f(h_i, m_{i+1}) = f(h'_i, m'_i)$$

در ادامه به معرفی طرح‌های دیویس-میر[۹] و ماتیاس-میر-اوسیاس[۱۰] می‌پردازیم که راه‌حل‌هایی برای رفع مشکل فوق می‌باشند.

[۹] Davies–Meyer
[۱۰] Matyas–Meyer–Oseas

۲.۴ طرح Davies-Meyer



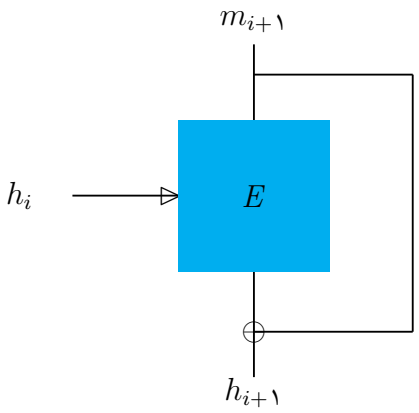
شکل ۱۷-۳: طرح دیویس-میر

محاسبه تابع فشرده‌ساز در این طرح به صورت زیر است.

$$h_{i+1} = f(h_i, m_{i+1}) = E_{m_{i+1}}(h_i) \oplus h_i$$

- در این طرح می‌توان از هر رمز قالب دلخواه استفاده نمود.
- اگر E یک رمزقالبی ایده‌آل باشد آنگاه یافتن برخورد نیاز به محاسبه $۲^{n/۲}$ پیام ورودی دارد.

۳.۴ طرح Matyas-Meyer-Oseas



شکل ۱۷-۴: طرح ماتیاس-میر-اوسیاس

برای محاسبه تابع فشرده‌ساز در این طرح از رابطه زیر استفاده می‌شود.

$$h_{i+1} = f(h_i, m_{i+1}) = E_{h_i}(m_{i+1}) \oplus m_{i+1}$$

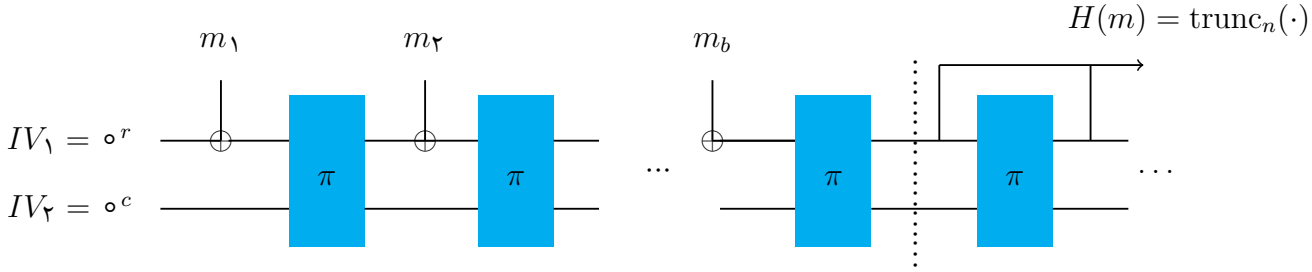
- در این طرح از رمز قالبی استفاده شود که طول قالب ورودی و طول کلید آن یکسان باشد.
- اگر E یک رمزقالبی ایده‌آل باشد آنگاه یافتن برخورد نیاز به محاسبه $۲^{n/۲}$ پیام ورودی دارد.

۵ طراحی توابع چکیده‌ساز در ساختار اسفنجی

برای ساخت یک تابع چکیده‌ساز با طول ورودی دلخواه و طول خروجی مطلوب در ساختار اسفنجی از یک ساختار تکرار شونده برمبنای یک جایگشت[۱۱] شبه‌تصادفی استفاده می‌شود. در شکل فوق π یک جایگشت روی $\{۰,۱\}^{r+c}$ است که r نرخ بیت[۱۲] و c ظرفیت[۱۳] نامیده می‌شود. پس از اعمال پدینگ روی پیام ورودی m (با اضافه کردن یک بیت یک و به تعداد لازم بیت صفر) و تقسیم آن به b قالب r بیتی $m_۱, m_۲, ..., m_b$، مقدار چکیده مطابق شکل ۱۷-۵ محاسبه می‌شود. ابتدا قالب‌های پیام پد شده اصطلاحاً توسط جایگشت جذب می‌شوند. سپس با اعمال مجدد جایگشت به تعداد لازم، چکیده پیام به طول مورد نیاز تولید می‌شود.

$$
\begin{aligned}
(x_۰, y_۰) &= IV = (۰^r, ۰^c) \\
(x_{i+۱}, y_{i+۱}) &= \pi(x_i \oplus m_{i+۱}, y_i) \quad i = ۰, ۱, \cdots, b-۱ \\
(x_{i+۱}, y_{i+۱}) &= \pi(x_i, y_i) \quad i = b, b+۱, \cdots \\
H(m) &= \mathrm{trunc}_n(x_b || x_{b+۱} || \cdots)
\end{aligned}
$$

تابع $\mathrm{trunc}_n(\cdot)$ فقط n اول دنباله‌ی ورودی را در نظر می‌گیرد.



شکل ۱۷-۵: ساختار اسفنجی

- درصورتی که π یک جایگشت ایده‌آل باشد آنگاه مشروط بر انتخاب مناسب پارامتر ظرفیت c تابع چکیده‌ساز H ایده‌آل خواهد بود.
- پیچیدگی هر حمله‌ای به ساختار اسفنجی محدود به پیچیدگی بهترین حمله به یک تابع چکیده‌ساز ایده‌آل و $۲^{c/۲}$ است.
- بنابراین اگر $n > c$ باشد، بهترین حمله برخورد دارای پیچیدگی $۲^{c/۲}$ خواهد بود (چنین حمله‌ای را بیابید).

[۱۱] permutation
[۱۲] bit-rate
[۱۳] capacity

۶ توابع چکیده‌ساز مهم

- **SHA-1:** این الگوریتم توسط NSA[۱۴] در ساختار مرکل-دمگارد، طراحی شده است و در سال ۱۹۹۵ توسط NIST[۱۵] انتشار یافت. طول خروجی این الگوریتم ۱۶۰ بیت وطول قالب آن ۵۱۲ بیت است. در سال ۲۰۰۵ نشان داده شد که با محاسبه 2^{69} مقدار ورودی می‌توان برخوردی برای این الگوریتم پیدا کرد. با وجود الگوریتم‌های اندکی بهتر هنوز برخوردی برای این تابع پیدا نشده است.

  پیش از سال ۲۰۱۰، SHA-1 کاربردهای گسترده‌ای در امضای دیجیتال DSS و پروتکل‌های SSL، TLS و PGP داشته است. در حال حاضر در بعضی از موارد SHA-1 با SHA-2 جایگزین گردیده است.

- **SHA-2:** مؤسسه NIST چهار تابع چکیده‌ساز که بر اساس طول خروجی شان نامگذاری شده بودند به نام های SHA-512، SHA-384، SHA-256 و SHA-224 معروف به خانواده SHA-2 را معرفی کرد. این خانواده از لحاظ ساختاری مشابه SHA-1 هستند. اما به طور کلی این خانواده با استقبال گسترده‌ای روبرو نشد.

- **MD5:** این الگوریتم در سال ۱۹۹۲ توسط ران ریوست[۱۶] در ساختار مرکل-دمگارد با طول قالب ۵۱۲ بیت وطول خروجی ۱۲۸ بیت طراحی گردید. حمله برخورد به تابع فشرده‌ساز این الگوریتم در سال ۱۹۹۶ ارایه شده است.

- **SHA-3:** موسسه NIST درسال ۲۰۰۷ مسابقه طراحی الگوریتم درهم ساز، با عنوان SHA-3 را برگزار نمود. از میان ۵۱ کاندیدای اولیه نهایتاً الگوریتم Keccak که در ساختار اسفنجی، طراحی شده بود به عنوان الگوریتم برتر انتخاب گردید و توسط NIST استاندارد شده است.

[۱۴] National Security Agency
[۱۵] National Institute of Standards and Technology
[۱۶] Ron Rivest