Shahid Beheshti University

رمزنگاری

هادی سلیمانی

پژوهشکده فضای مجازی دانشگاه شهید بهشتی

مجوز استفاده و نشر

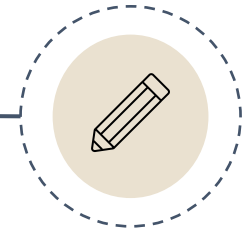
- اجازه‌ی ایجاد نسخه‌های دیجیتالی جدید براساس بخشی یا تمام مطالب این اسلاید بدون پرداخت هزینه اعطا می‌شود، مشروط بر این‌که:
  - فقط به‌منظور و در راستای استفاده‌ی آموزشی (شخصی و یا کلاسی) ساخته شده باشند و برای کسب هرگونه سود و یا مزیت تجاری استفاده نشوند.
  - نسخه‌های جدید حاوی ارجاع مستقیم به نام تهیه‌کننده اسلاید (هادی سلیمانی) و محل کار وی (پژوهشکده فضای مجازی دانشگاه شهید بهشتی) باشند.
- مجموعه‌ی حاضر براساس نظرات ارزشمند دانشجویان (سابق) دانشگاه شهید بهشتی و همکاران محترم تهیه شده است که از تمام آن‌ها قدردانی می‌شود؛

  (به‌خصوص خانم‌ها **سارا زارعی** و **فاطمه عزیزی** نقش مهمی را در تهیه نسخه‌ی نهایی بر عهده داشته‌اند. خانم مهندس زارعی علاوه بر کمک در آماده‌سازی نسخه‌ی فعلی اسلایدها، در تصحیح اشتباهات نسخه‌ی قبلی و همچنین تکمیل و بازتعریف محتوای درس‌ها بسیار تاثیرگذار بوده‌اند).
- برای مشاهده‌ی اسلایدها و ویدئوهای تدریس این درس به آدرس زیر مراجعه فرمایید:

http://facultymembers.sbu.ac.ir/h_soleimany/cryptography-course/

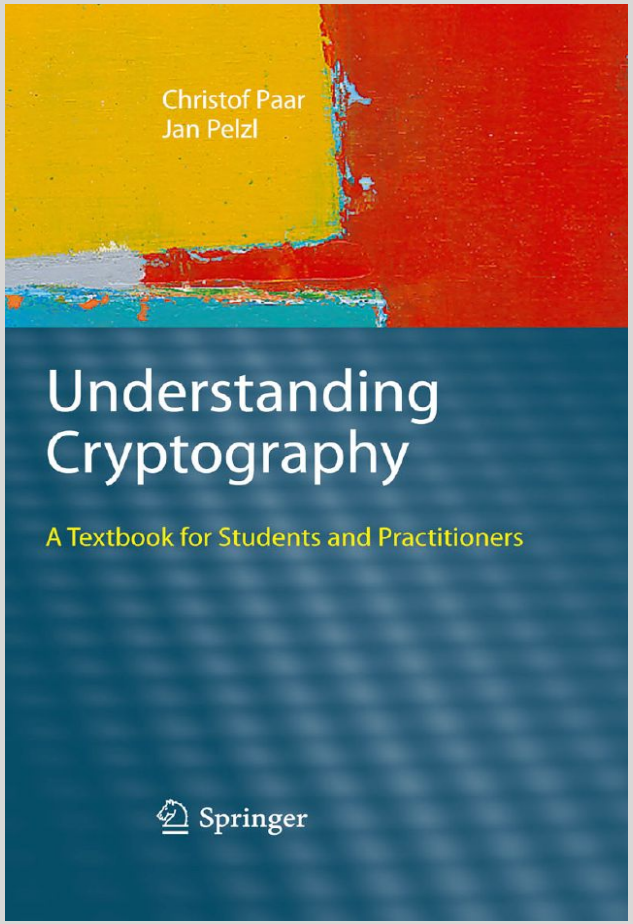
درس چهاردهم

تابع چکیده‌ساز



معرفی مـرجع

تابع چکیده‌ساز

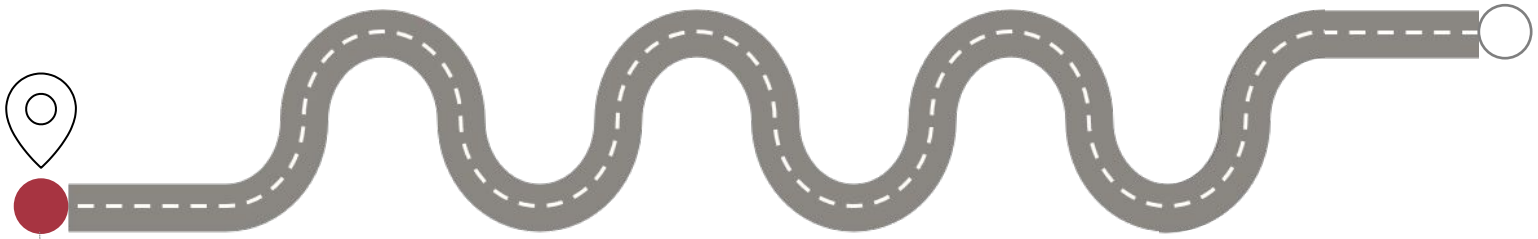


Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.

مثال‌ها و تصاویر این بخش از درس از کتاب مرجع است (در غیر این صورت مرجع شکل ذکر شده است).

فهرست عناوین درس

تابع چکیده‌ساز

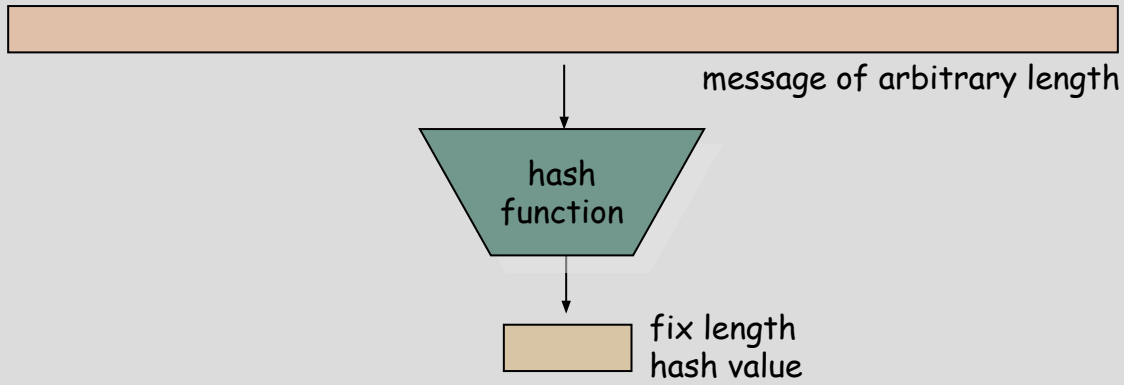


مقدمه

تابع چکیده‌ساز

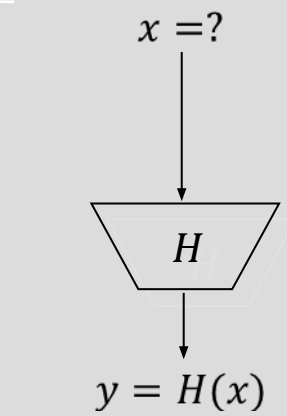
(Hash Function)

- یک مقدار ورودی با طول دلخواه را به یک مقدار خروجی با طول ثابت تبدیل می‌کند.
- تابع چکیده‌سازی مناسب است که مقادیر خروجی آن به ازای ورودی‌های مختلف، در عمل متفاوت شوند.
- به عبارتی می‌توان گفت که خروجی یک تابع چکیده‌ساز خوب باید مانند اثر انگشت افراد منحصربه‌فرد باشد.
- همچنین تابع چکیده‌ساز باید به‌نحوی باشد که محاسبه‌ی آن به لحاظ زمانی ساده باشد.

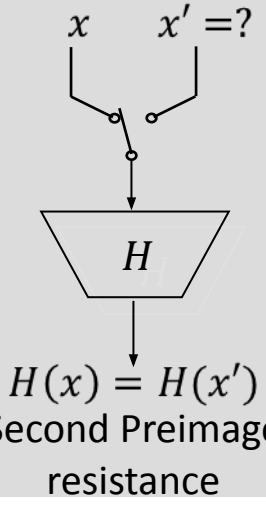


تابع چکیده‌ساز امن

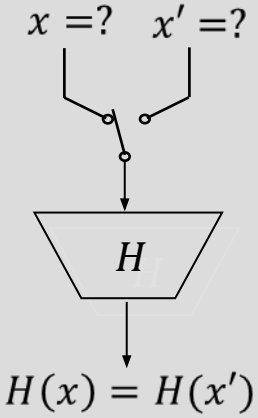
- در کاربردهای رمزنگاری، تابع چکیده‌سازی مناسب است که ویژگی‌های زیر را داشته باشد:
- مقاومت در برابر پیش‌تصویر (Preimage Resistance)
  - برای یک مقدار y داده شده، نتوان یک مقدار x پیدا کرد به نحوی که $H(x) = y$.
- مقاومت در برابر پیش تصویر دوم (Second Preimage Resistance)
  - با داشتن مقدار x کسی نتواند مقدار متفاوت x' را به نحوی پیدا کند که $H(x) = H(x')$.
- مقاومت در برابر برخورد (Collision Resistance)
  - نتوان مقادیر متفاوت x و x' را پیدا کرد به نحوی که $H(x) = H(x')$.



Preimage resistance



Second Preimage resistance



Collision resistance

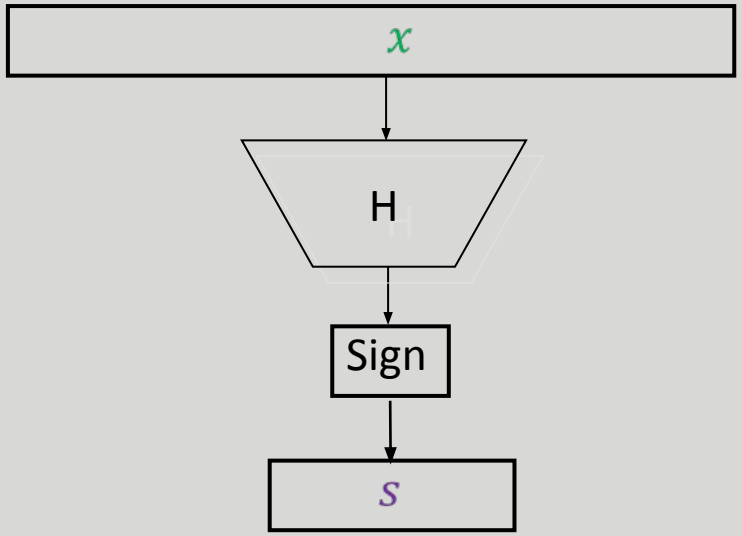
برخی کاربردهای
تابع چکیده‌ساز

مقدمه

استفاده از تابع چکیده‌ساز

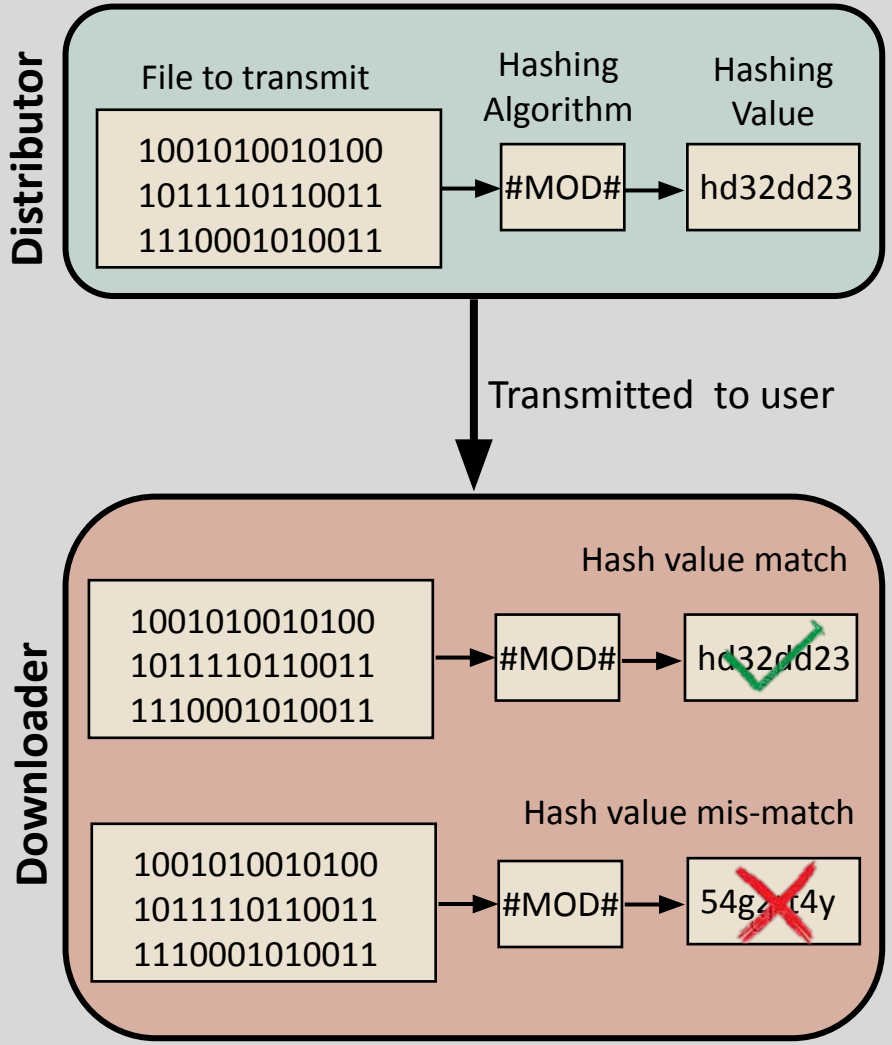
یادآوری

- برای مقابله با برخی از حملات جعل، می‌توان از تابع چکیده‌ساز امن استفاده کرد.
- استفاده از تابع چکیده‌ساز نه تنها امنیت را افزایش می‌دهد بلکه کارایی را نیز افزایش می‌دهد:
  1. طول امضا کاهش می‌یابد.
  2. به تعداد عملیات‌های امضای کمتری نیاز است و در نتیجه سرعت (به مراتب) بیشتر می‌شود.



جامعیت پیام

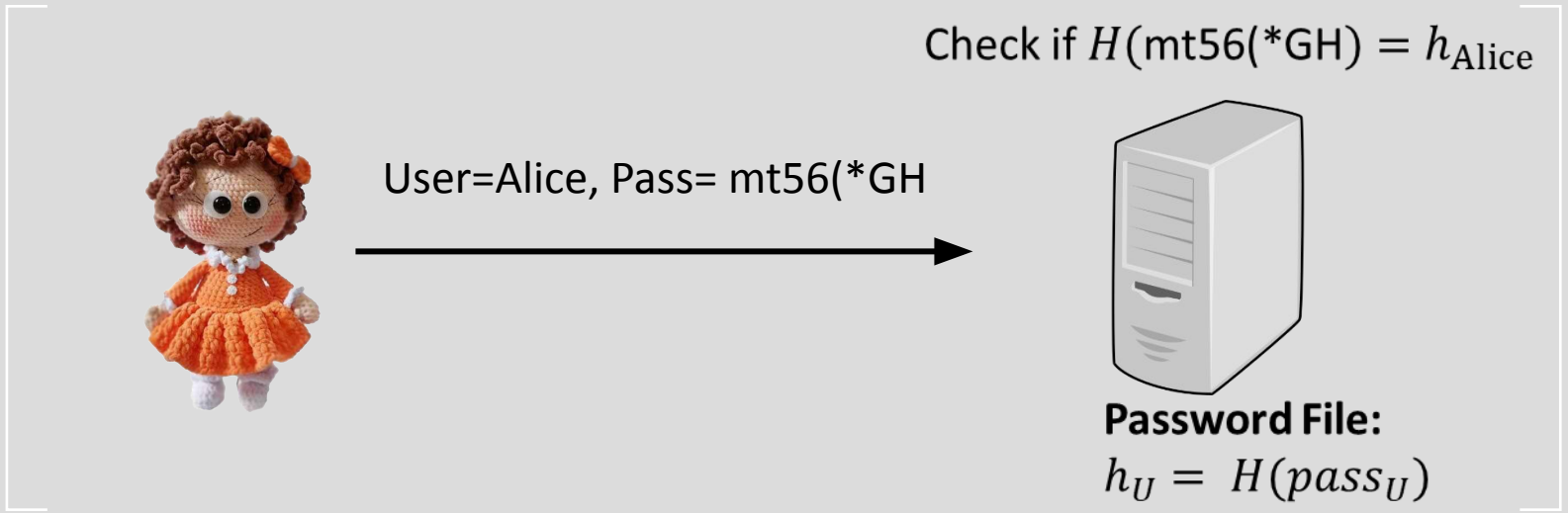
- پیدا کردن تصادم (برخورد) برای یک تابع چکیده‌ساز امن کار دشواری است.
- بر همین اساس، اگر $H(x) = H(y)$ باشد، می‌توان با اطمینان بالایی نتیجه گرفت که $x = y$ بوده است.
- برای ارزیابی این‌که آیا یک فایل تغییر پیدا کرده است یا خیر، کافی است که مقدار چکیده‌ی آن را (که به مراتب کوچک‌تر است) ذخیره کرد.
- به عنوان مثال برای اصالت سنجی فایل‌های بروزرسانی نرم‌افزارها و ...، می‌توان از این خاصیت توابع چکیده‌ساز استفاده کرد.



Figure's source: Wikipedia

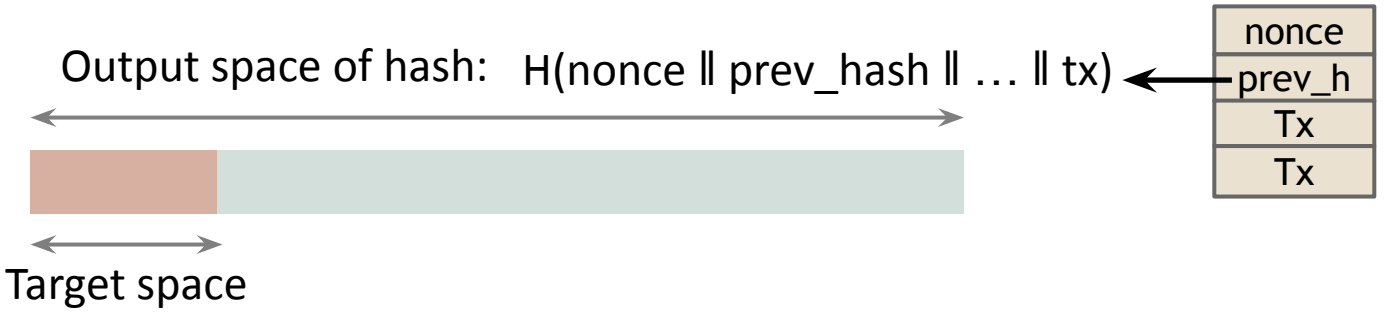
محافظت از گذرواژه‌ها

- اگر گذرواژه‌ی افراد به صورت کامل در سرور ذخیره شود، در صورت دزدیده شدن فایل، اطلاعات تمامی کاربران افشا می‌شود!
- یک راهکار مقابله این است که سرور به جای گذرواژه، چکیده‌ی گذرواژه‌ی هر کاربر را ذخیره کند.
- وقتی یک کاربر گذرواژه خود را ارسال می‌کند، مقدار چکیده‌ی گذرواژه محاسبه شده و با مقدار ذخیره شده مقایسه می‌شود.



سنجش توان محاسباتی

- فرض کنید که قرار است به‌ازای یک داده‌ی تصادفی r، یک مقدار $nonce$ پیدا شود به گونه‌ای که $H(nonce||r)$ در ۷۰ بیت اول 0 باشد.
- اگر تابع چکیده‌ساز امن باشد، تنها از راه تکرار محاسبات می‌توان مقدار $nonce$ مناسب را پیدا کرد.
- به طور متوسط انتظار می‌رود بتوان با 2^{70} محاسبه‌ی مقادیر مختلف $nonce$ به پاسخ رسید.
- پس اگر فردی پاسخی برای این سوال پیدا کند، احتمالا دارای یک توان محاسباتی از مرتبه 2^{70} بوده است!
- به این ویژگی اصطلاحا Puzzle-friendly گفته می‌شود که در بیت‌کوین نیز از آن برای سنجش توان محاسباتی افراد استفاده می‌شود.



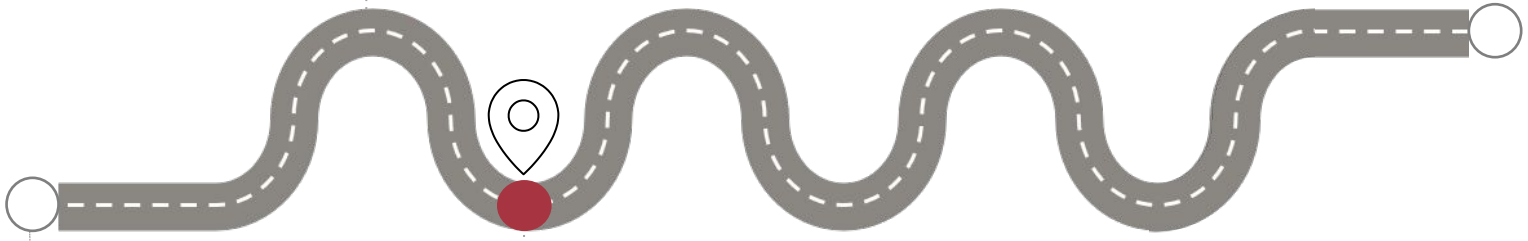
Figure's source: "Intro to Crypto and Cryptocurrencies" by Joseph Bonneau, et. al.

طرح تعهد

(Commitment Scheme)

- در دنیای فیزیکی، برای برگزاری یک مناقصه یا مزایده راهکار ساده‌ای وجود دارد:
  1. هر شرکت‌کننده عدد خود را درون پاکت سربسته و مهرشده قرار داده و ارسال می‌کند.
  2. سپس در روز اعلام نتایج، پاکت‌ها با حضور شرکت‌کنندگان باز می‌شوند.
- برای دنیای مجازی نیز راهکارهایی براساس تابع چکیده‌ساز وجود دارد.
- مثلا می‌توان مقدار ثابت k و یک تابع چکیده‌ساز H را اعلام کرد و از شرکت‌کنندگان خواست که مبلغ پیشنهاد خود (x) را به صورت زیر ارسال کنند:
  1. ابتدا یک عدد تصادفی r تولید کنند.
  2. مقدار چکیده‌ی $H(x||k||r)$ را محاسبه و اعلام کنند.
- پس از دریافت تمامی پیشنهادها، از افراد خواسته می‌شود که مقادیر x و r خود را اعلام کنند.

برخی
کاربردهای تابع
چکیده‌ساز

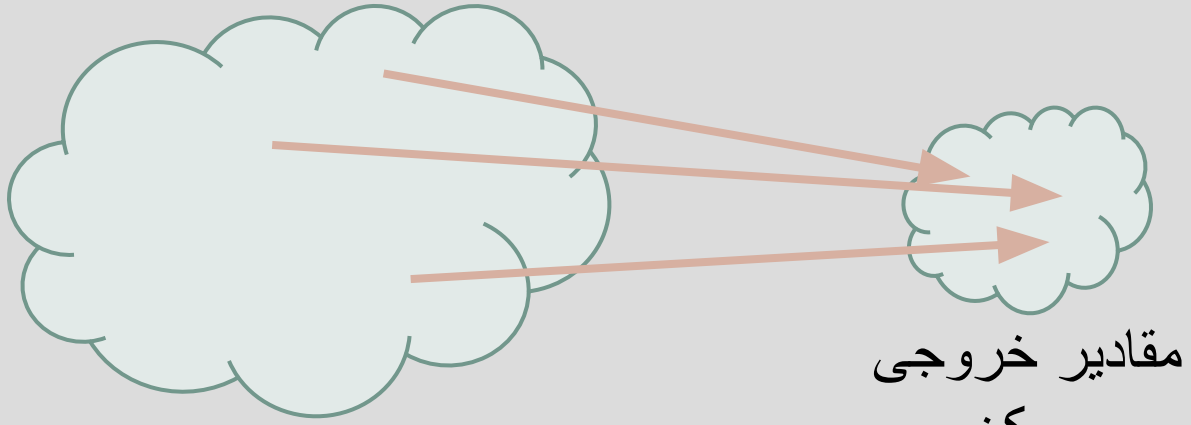


مقدمه

کران امنیتی
تابع
چکیده‌ساز

وجود تصادم

- با توجه به این‌که اندازه‌ی فضای ورودی بسیار بزرگ‌تر از اندازه‌ی فضای خروجی است، تصادم همیشه وجود دارد!
- تابع چکیده‌ساز باید به گونه‌ای باشد که **در عمل** نتوان برای آن تصادم پیدا کرد.
- سوالی که پیش می‌آید این است که حداکثر امنیت محاسباتی که یک تابع چکیده‌ساز می‌تواند داشته باشد، چقدر است؟



Figure's source: "Intro to Crypto and Cryptocurrencies" by Joseph Bonneau, et. al.

مسئله‌ی روز تولد

(Birthday Problem)

- شرح مسئله: احتمال این‌که در بین t نفری که به صورت تصادفی انتخاب شده‌اند، حداقل دو نفری در یک روز به دنیا آمده باشند، چقدر است؟

$$\Pr(\text{no collision among 2 people}) = \left(\frac{364}{365}\right)$$

$$\Pr(\text{no collision among 3 people}) = \left(\frac{364}{365}\right).\left(\frac{363}{365}\right)$$

$$\Pr(\text{no collision among } t \text{ people}) = \left(\frac{364}{365}\right).\left(\frac{363}{365}\right)\ldots\left(\frac{365-(t-1)}{365}\right)$$

$$\Pr(\text{at least one collision amont } t \text{ people}) = 1 - \Pr(\text{no collision among } t \text{ people})$$

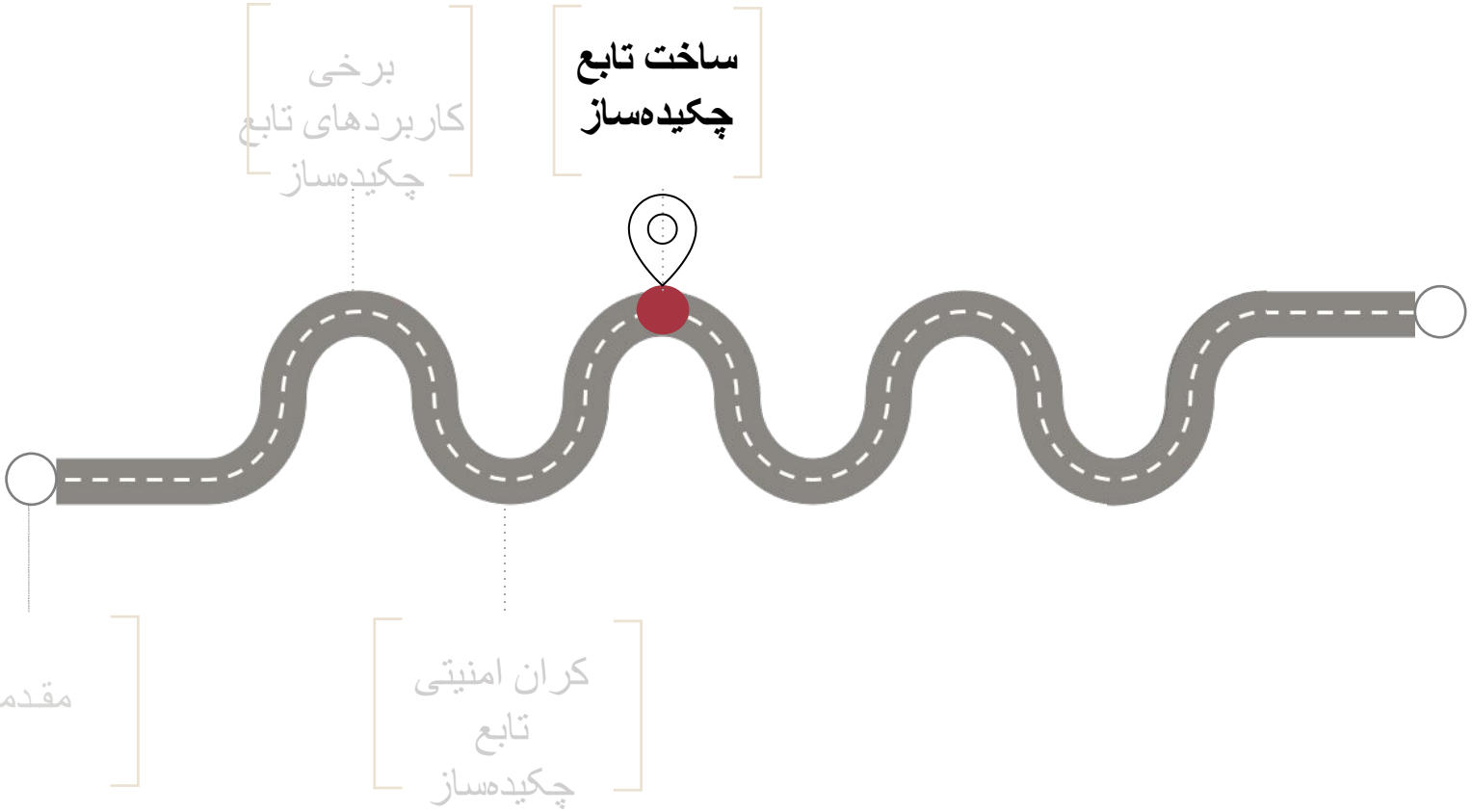
- به ازای $t = 23$، با احتمال بالای ۵۰ درصد، دو نفر وجود دارند که در یک روز متولد شده باشند.

پیدا کردن تصادم

- در حالت کلی اگر فضای حالت‌های ممکن m باشد، با احتمال بالای ۵۰ درصد، $\sqrt{m}$ مقدار تصادفی تولید شده دارای تصادم هستند.
  - برای مشاهده‌ی اثبات به کتاب مرجع، بخش ۱۱.۲.۳ مراجع کنید!
- با تولید حدود $2^{n/2}$ مقدار تصادفی و محاسبه‌ی چکیده‌ی آنها، می‌توان با احتمال بالایی برای یک تابع چکیده‌ساز با طول خروجی n بیت یک تصادم پیدا کرد.
- بنابراین امنیت هیچ تابع چکیده‌سازی نمی‌تواند در مقابل تصادم بیشتر از $2^{n/2}$ باشد.

یافتن پیش تصویر (دوم)

- فرض کنید که هیچ رابطه‌ی آماری قابل استفاده‌ای بین ورودی و خروجی تابع چکیده‌ساز H با خروجی n وجود نداشته باشد.
- احتمال آن‌که $H(x)$ برابر یک مقدار ثابت و مشخص n بیتی شود برابر با 2^{-n} است.
- تعداد محاسبات لازم در این حالت برای یافتن پیش تصویر (دوم) برای یک تابع چکیده‌ساز با طول خروجی n بیت، از مرتبه‌ی 2^n است.
- بنابراین حداکثر امنیت ممکن برای یک تابع چکیده‌ساز با طول خروجی n بیت، 2^n است.


برخی
کاربردهای تابع
چکیده‌ساز
ساخت تابع
چکیده‌ساز
مقدمه
کران امنیتی
تابع
چکیده‌ساز

ساخت تابع چکیده‌ساز

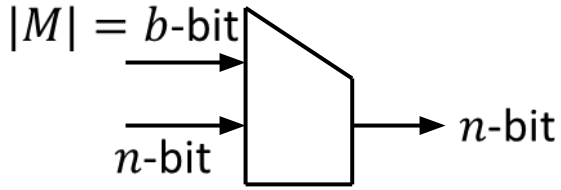
- طراحی یک تابع چکیده‌ساز بدون ویژگی‌های امنیتی توصیف شده، ساده است!
- اما طراحی یک تابع چکیده‌ساز امن و کارآمد است که چالش‌برانگیز است، چراکه تابع چکیده‌ساز هیچ پارامتر مخفی‌ای (مانند کلید) ندارد.
- به همین علت است که تعداد قابل توجهی از طرح‌های ارائه شده تاکنون شکسته شده‌اند.

رویکرد معمول

- رویکرد معمول این است که ابتدا یک تابع فشرده‌ساز با طول ورودی ثابت ساخته می‌شود و سپس به صورت متناوب مورد استفاده قرار می‌گیرد.
- تعریف تابع فشرده‌ساز (Compression Function):

$$f: \{0,1\}^{b+n} \rightarrow \{0,1\}^{n}$$

- تابع فشرده‌ساز باید در مقابل تصادم امن باشد.



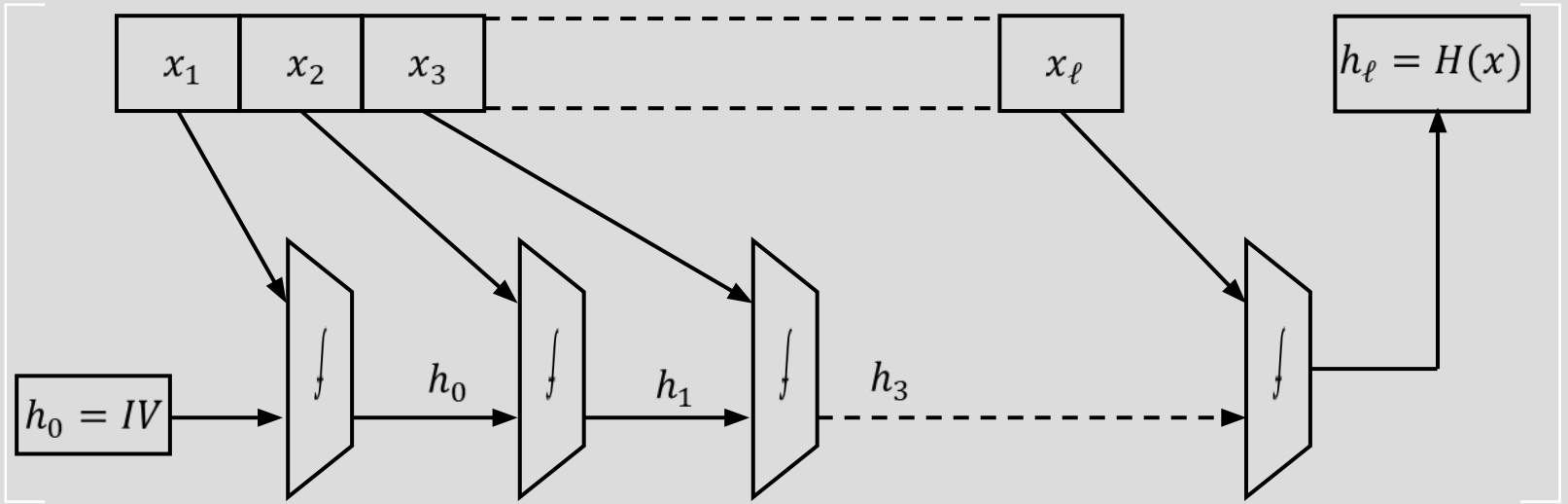
ساختار مرکل ـ دمگارد

(Merkle–Damgård Construction)

- Padding: پیام x را با اضافه کردن مقدار $|x|$ و همچنین تعدادی بیت 0 گسترش می‌دهیم به گونه‌ای که طول پیام مضرب b شود.
- پیام x را به قالب‌های b بیتی $(x_1, x_2, \ldots, x_\ell)$ تجزیه کرده و چکیده‌ی پیام را به صورت زیر محاسبه می‌کنیم:

$$h_i = f(x_i, h_{i-1}), \text{ where } h_0 = IV$$

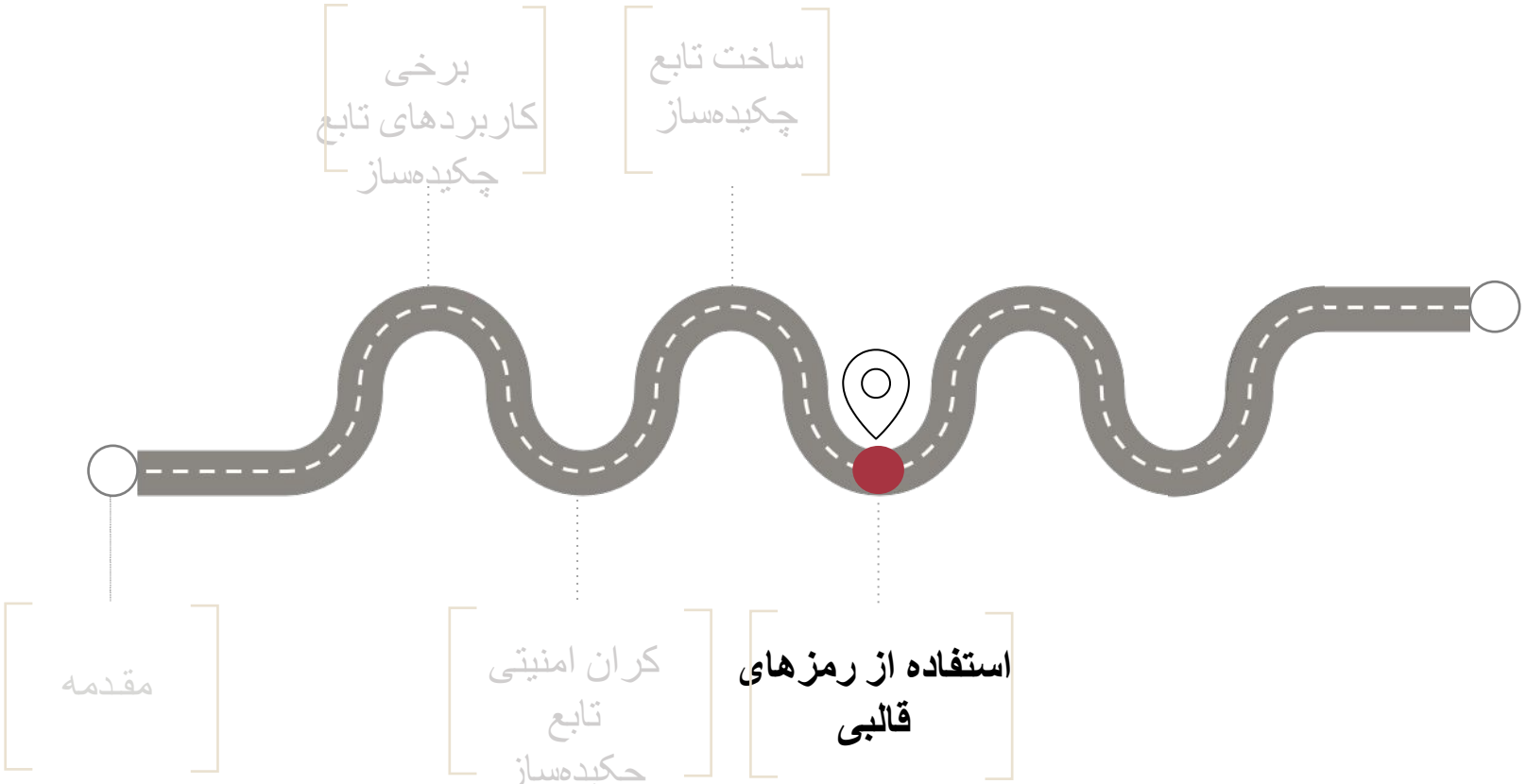
- قضیه مرکل – دمگارد: اگر f (شبه) برخوردمقاوم باشد آن‌گاه H برخوردمقاوم است.



مزایای ساختارهای تکرارشونده

1. پیام ممکن است در قالب packetهای کوچک بیاید. بنابراین قبل از دریافت کامل پیام می‌توان فرآیند محاسبه‌ی چکیده را شروع کرد.
2. مقدار حافظه‌ی مورد نیاز محدود است.
   - هنگامی که یک بلوک از پیام محاسبه شد، دیگر نیازی به نگهداری آن نیست.
3. تجزیه و تحلیل امنیتی ساده‌تر می‌شود.
   - می‌توان صرفا بر روی تحلیل امنیتی توابع کوچک‌تر تمرکز کرد.

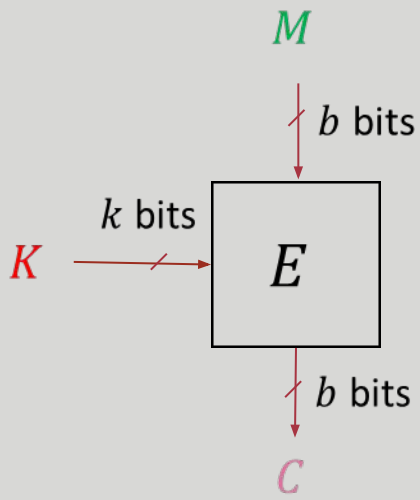
- به جز ساختار مرکل - دمگارد، ساختارهای تکرارشونده‌ی دیگری هم ارائه شده‌اند که با برخی از آن‌ها آشنا خواهیم شد.
- برای padding می‌توان از روش‌های مختلفی استفاده کرد.
- برخی توابع فشرده‌ساز اختصاصی هستند و برخی براساس رمزهای قالبی ارائه شده‌اند.


برخی کاربردهای تابع چکیده‌ساز
ساخت تابع چکیده‌ساز
مقدمه
کران امنیتی تابع چکیده‌ساز
استفاده از رمزهای قالبی

رمز قالبی

یادآوری

- الگوریتم رمز قالبی E، یک متن اصلی b بیتی و یک کلید k بیتی (K) را به عنوان ورودی گرفته و به یک متن رمزشده‌ی b بیتی تبدیل می‌کند.
- بنابراین الگوریتم رمز قالبی، ماهیتا یک تابع فشرده‌ساز است.
- رمز قالبی را نمی‌توان به تنهایی به عنوان یک تابع فشرده‌ساز امن استفاده کرد. چرا؟!
- اما ساختارهایی وجود دارند که براساس آن‌ها می‌توان از روی یک رمز قالبی امن، یک تابع فشرده‌ساز امن ساخت.



$$E: \{0,1\}^b \times \{0,1\}^k \rightarrow \{0,1\}^b$$

ساخت تابع چکیده‌ساز با استفاده از رمز قالبی

ساختار Miyaguchi-Preneel

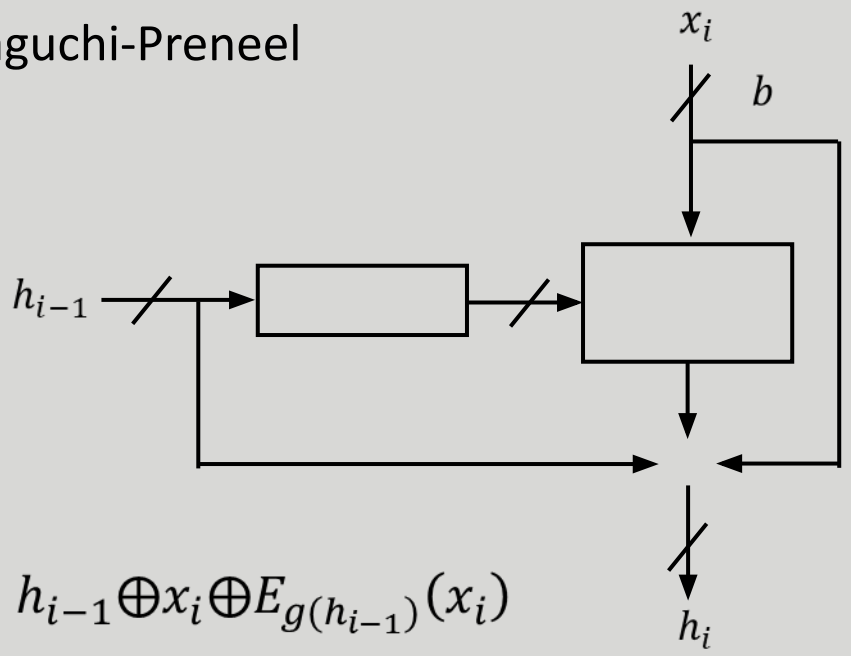
1. b بیت h_{i-1} توسط یک نگاشت (g) به k بیت تبدیل می‌شود.
2. مقدار x_i تحت کلید $g(h_{i-1})$ رمز می‌شود.
3. مقدار $h_i = h_{i-1} \oplus x_i \oplus E_{g(h_{i-1})}(x_i)$ محاسبه می‌شود.

ساختار Davies-Meyer:

1. h_{i-1} تحت کلید x_i رمز می‌شود.
2. مقدار $h_i = h_{i-1} \oplus E_{x_i}(h_{i-1})$ محاسبه می‌شود.

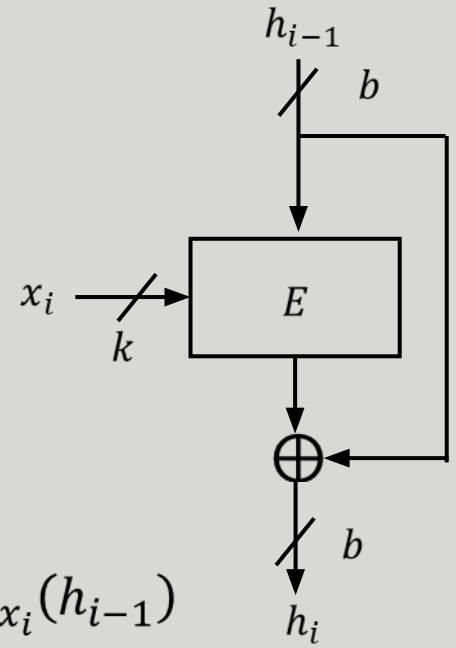
- امنیت قابل اثبات این طرح‌ها در مقابل پیش‌تصویر (دوم) برابر 2^b و در برابر تصادم $2^{b/2}$ است.

Miyaguchi-Preneel



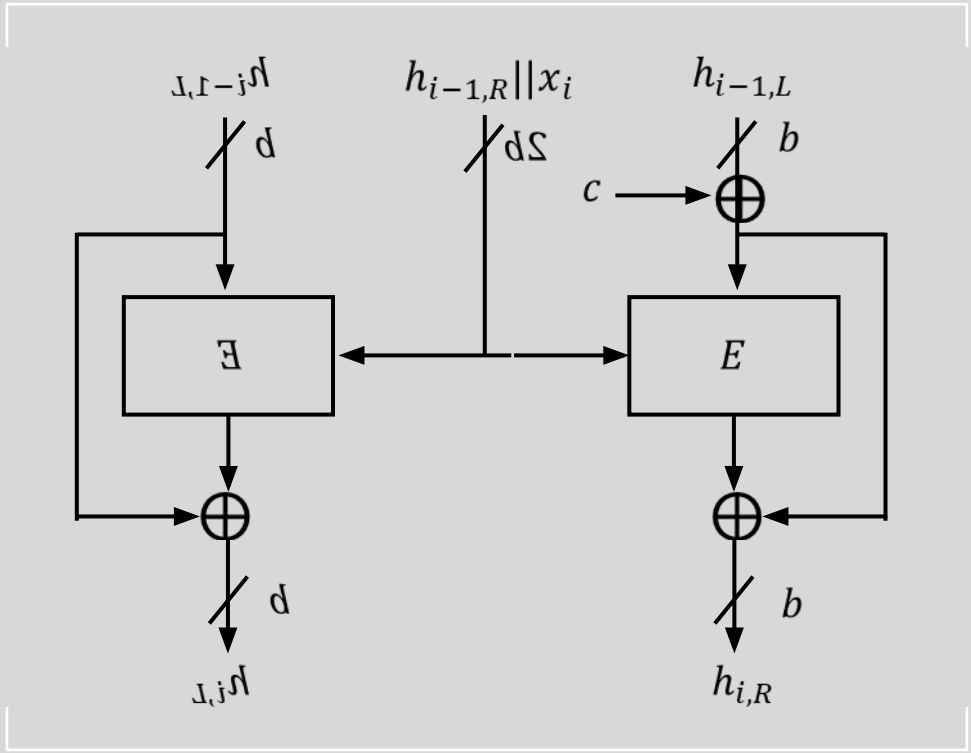
$$h_i = h_{i-1} \oplus x_i \oplus E_{g(h_{i-1})}(x_i)$$

Davies-Meyer

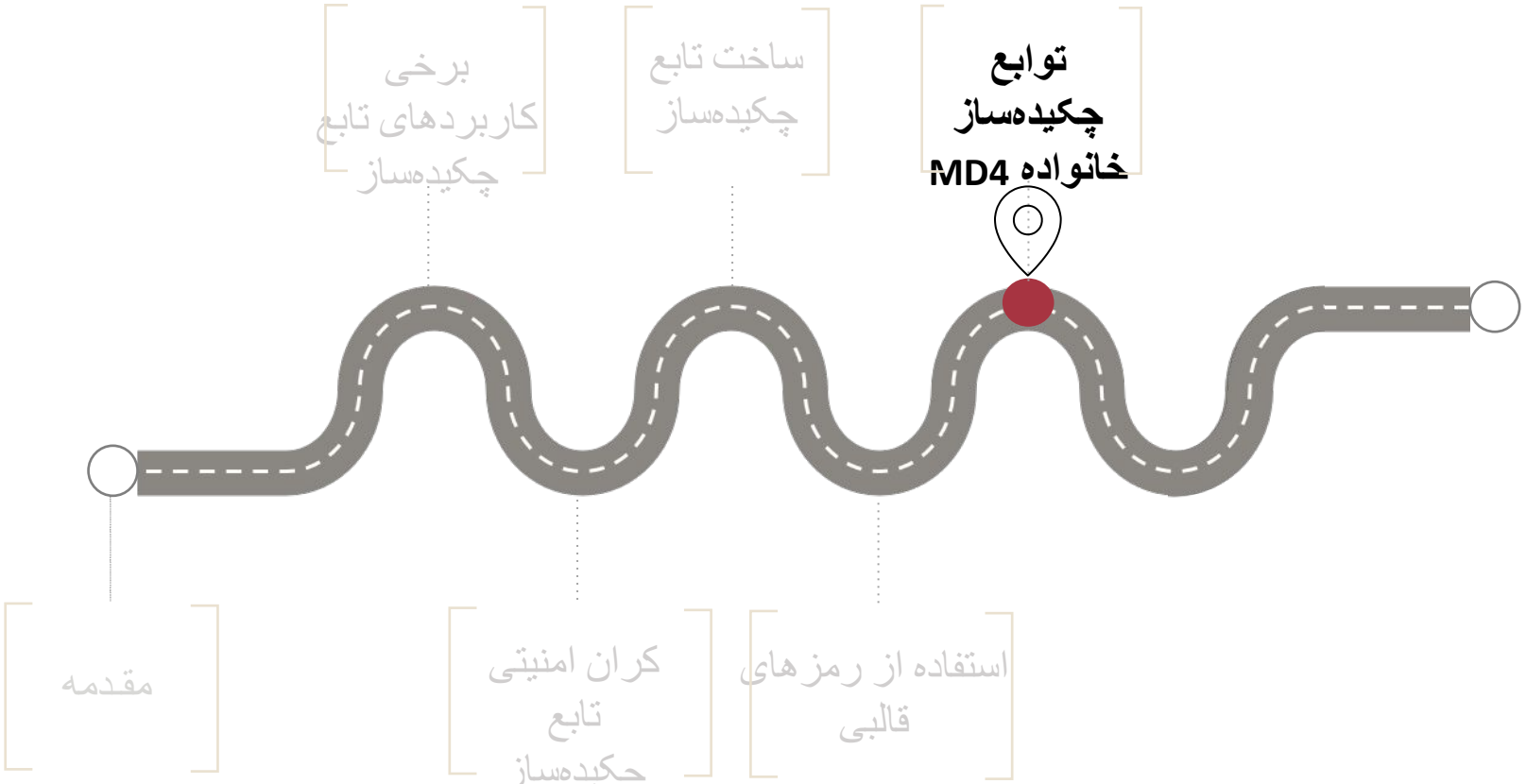


$$h_i = h_{i-1} \oplus E_{x_i}(h_{i-1})$$

ساختار Hirose



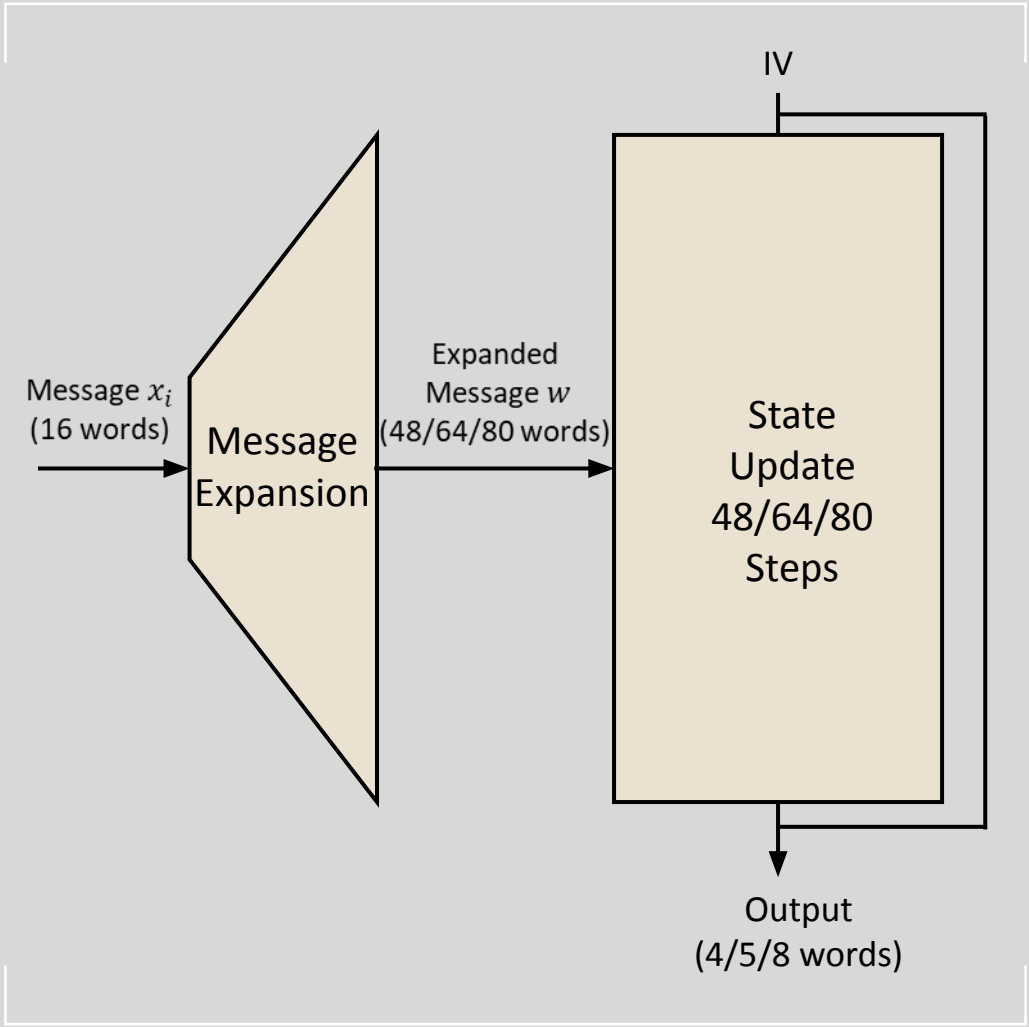
- در ساختارهای معرفی شده در اسلاید قبل، اگر به امنیت در مقابل تصادم نیاز باشد، باید یک رمز قالبی با طول حداقل ۲۵۶ بیت استفاده شود.
- بنابر این نمی‌توان از رمزهای استاندار نظیر AES-128 استفاده کرد.
- یک راه‌کار دیگر استفاده از ساختار Hirose است که از یک رمز قالبی امن که طول کلید آن دو برابر طول قالب است استفاده می‌کند (همانند AES-256).
- b بیت سمت راست h_{i-1} (یعنی $h_{i-1,R}$) و b بیت x_i به عنوان کلید در نظر گرفته می‌شوند.
- b بیت سمت چپ h_{i-1} (یعنی $h_{i-1,L}$) یک بار به صورت عادی و یک بار پس از xor با مقدار غیرصفر c رمز می‌شود.


برخی
کاربردهای تابع
چکیده‌ساز
ساخت تابع
چکیده‌ساز
توابع
چکیده‌ساز
خانواده MD4
مقدمه
کران امنیتی
تابع
چکیده‌ساز
استفاده از رمزهای
قالبی

ساختار کلی

- MD4 یک تابع چکیده‌ساز است که توسط Ronald Rivest ارائه شد.
- این طراحی، الهام‌بخش طراحی توابع چکیده‌ساز دیگری شد که امروزه عنوان خانواده MD4 شناخته می‌شوند.
- تمامی توابع چکیده‌ساز خانواده MD4 مبتنی بر ساختار مرکل – دمگارد هستند و اشتراکاتی در تابع فشرده‌ساز دارند:

1. قالب پیام x_i (۱۶ کلمه ۳۲ بیتی)، بسط داده می‌شود (۴۸ یا ۶۴ و یا ۸۰ کلمه).
2. تابع فشرده‌ساز با IV مقداردهی اولیه شده، سپس در مراحل متوالی توسط قالب‌های بسط‌یافته از پیام به‌روز می‌شود.
3. خروجی نهایی با استفاده از حالت نهایی و IV (معمولاً با استفاده از یک جمع ساده) حاصل می‌شود.



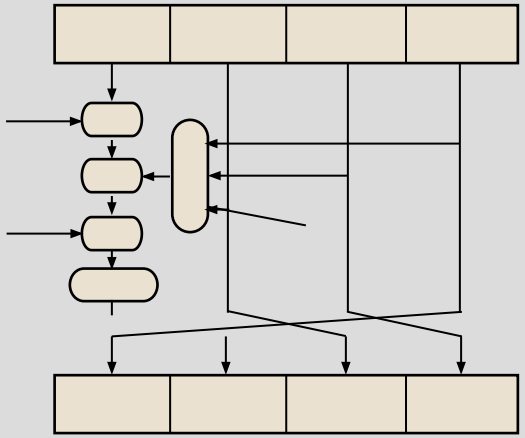
اعضای خانواده MD4

- نسخه‌ی قوی‌تر MD4 با نام MD5 در سال ۱۹۹۱ توسط Rivest ارائه شد که طول خروجی آن مانند MD4، ۱۲۸ بیت بود.
- با توجه به ضعف‌هایی که به‌مرور در MD5 مشاهده شد، NIST در سال ۱۹۹۳ طرحی با نام Secure Hash Algorithm (SHA) با ۱۶۰ بیت خروجی معرفی کرد که با نام SHA-0 شناخته می‌شود.
- NIST نسخه‌ی تقویت‌شده‌ی دیگری را نیز با طول خروجی ۱۶۰ بیت و با نام SHA-1 در سال ۱۹۹۵ ارائه کرد.
- کران امنیت SHA-0 و SHA-1 در مقابل تصادم 2^{80} است که برای کاربرد در برخی پروتکل‌ها کافی نیست.
- بر همین اساس NIST نسخه‌هایی با طول‌های ۲۲۴، ۲۵۶، ۳۸۴ و ۵۱۲ را معرفی کرد که با عناوین SHA-224، SHA-256، SHA-384 و SHA-512 شناخته می‌شوند.
- عموما به این چهار نسخه SHA-2 گفته می‌شود.

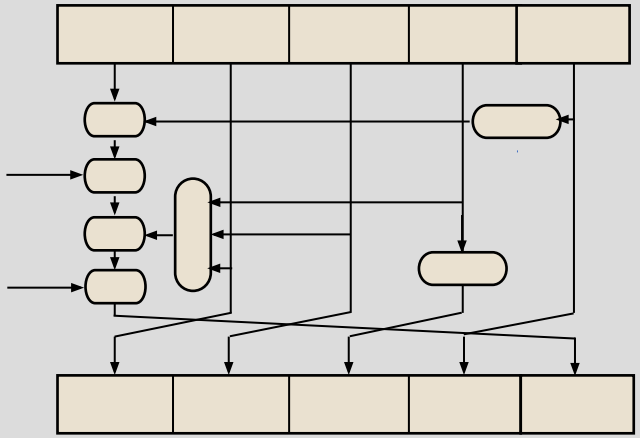
تابع به روزرسانی

- تابع به‌روزرسانی در توابع خانواده MD4 متفاوت است.

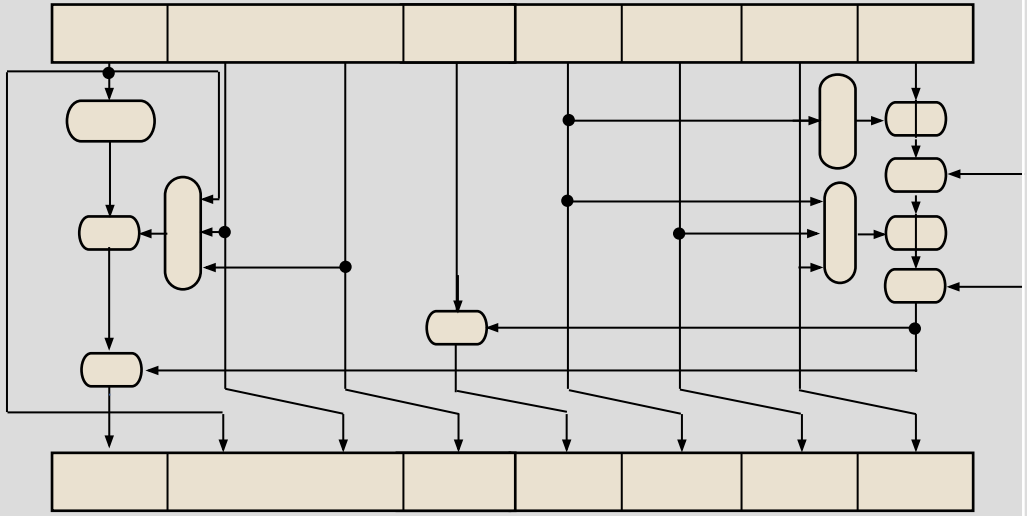
MD4



SHA/SHA-1



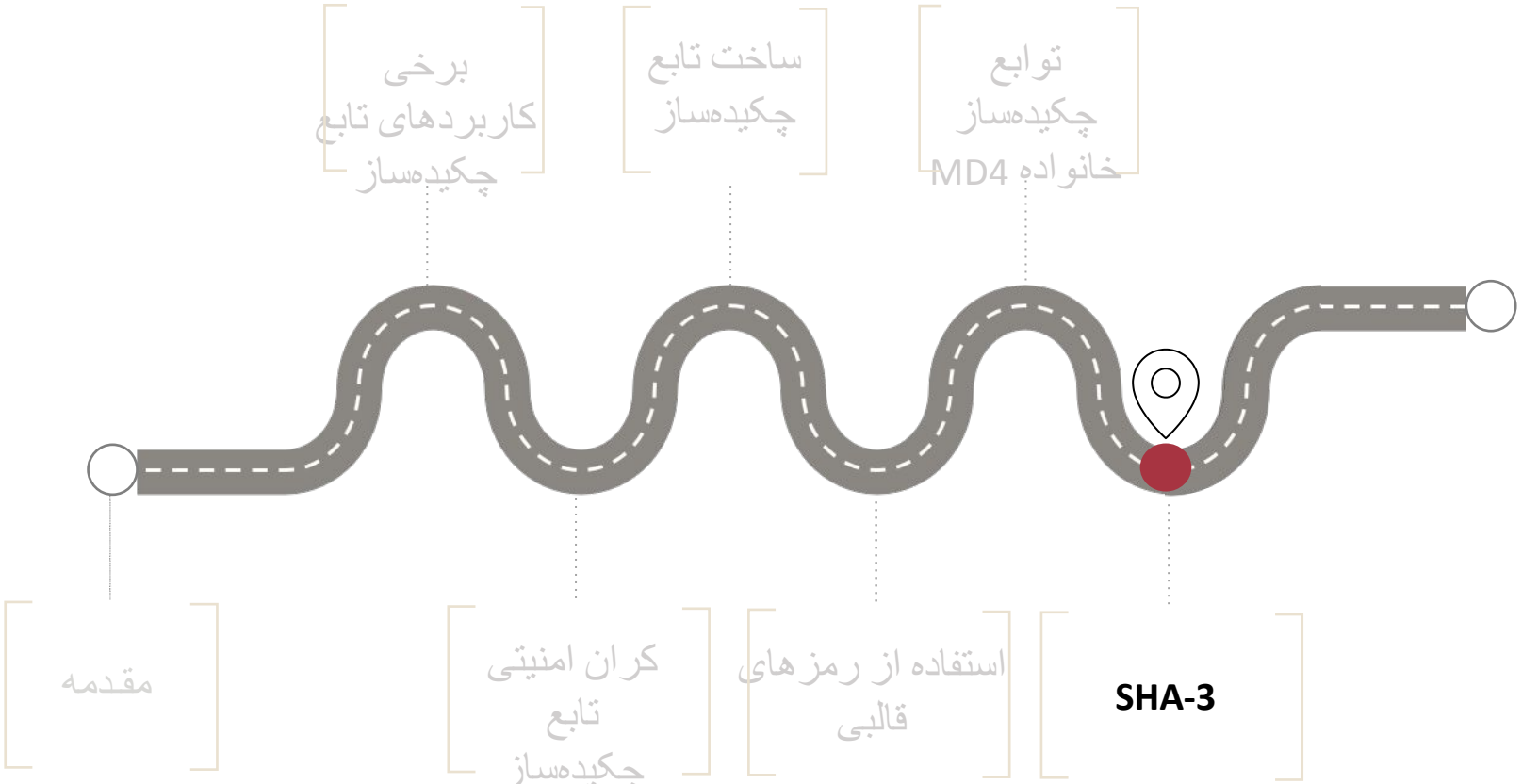
SHA-2 members



امنیت توابع خانواده MD4

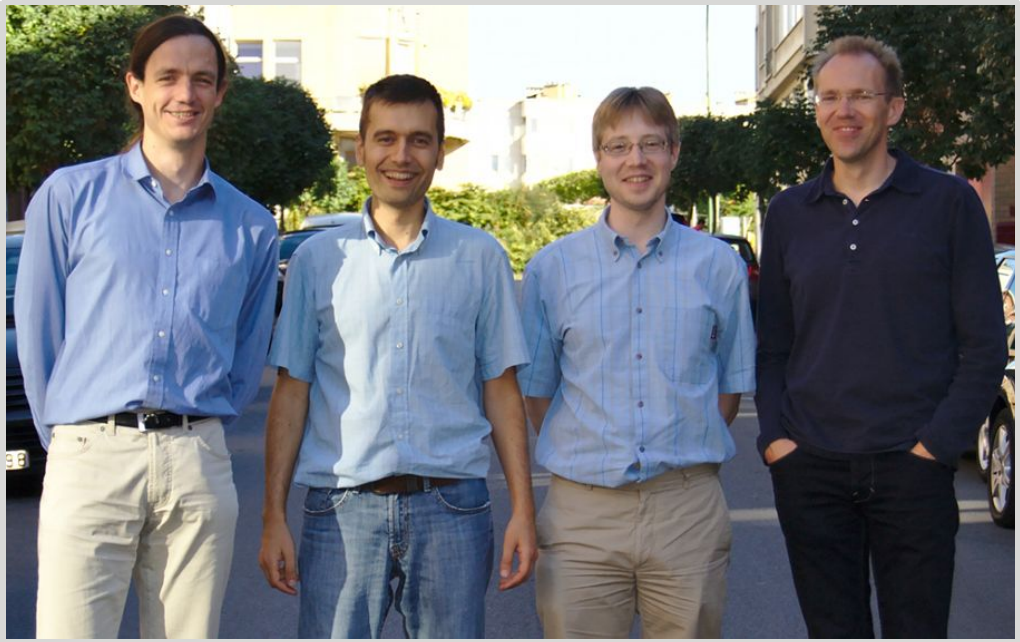
- در سال ۲۰۰۴، روش توسط وانگ و همکارنش برای یافتن تصادم ارائه شد که براساس آن تاکنون برای MD4، MD5 SHA-0 و SHA-1 تصادم پیدا شده است.
- البته باید دقت کرد که در برخی از کاربردها تنها امنیت در مقابل پیش‌تصویر نیاز است.

Algorithm		Output [bit]	Input [bit]	No of rounds	Collisions found
MD5		128	512	64	Yes
SHA-0		160	512	80	Yes
SHA-1		160	512	80	Yes
SHA-2	SHA-224	224	512	64	No
	SHA-256	256	512	64	No
	SHA-384	384	1024	80	No
	SHA-512	512	1024	80	No


برخی کاربردهای تابع چکیده‌ساز
ساخت تابع چکیده‌ساز
توابع چکیده‌ساز خانواده MD4
مقدمه
کران امنیتی تابع چکیده‌ساز
استفاده از رمزهای قالبی
SHA-3

مسابقه‌ی SHA-3

- بعد از حملات موفقیت‌آمیز وانگ و همکارانش روي SHA-0 و NIST، SHA-1 تصمیم گرفت تا مسابقه‌اي را براي انتخاب یک تابع چکیده‌ساز امن به اسم SHA-3 ترتیب دهد.
- NIST در سال 2007 اطلاعیه‌ی مسابقه SHA3 را منتشر کرد.
- سال ۲۰۰۸: ارسال ۶۴ طرح و پذیرفته شدن ۵۱ طرح در مسابقه.
- سال ۲۰۰۹: انتخاب ۱۴ طرح برای دور دوم (تعداد زیادی از طرح‌های دور اول شکسته شدند!).
- سال ۲۰۱۰: انتخاب ۵ فینالیست.
- سال ۲۰۱۱: انتخاب الگوریتم Keccak (با ساختار جدیدی به نام اسفنجی) به عنوان برنده‌ی مسابقه.

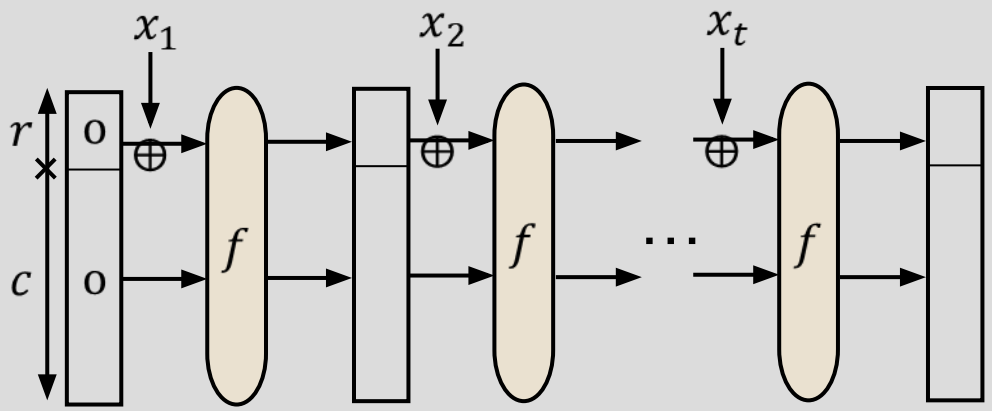


تیم طراحان:
Guido Bertoni, Joan Daemen and Gilles Van Assche, Michaël Peeters

ساختار اسفنجی

(Sponge Construction)

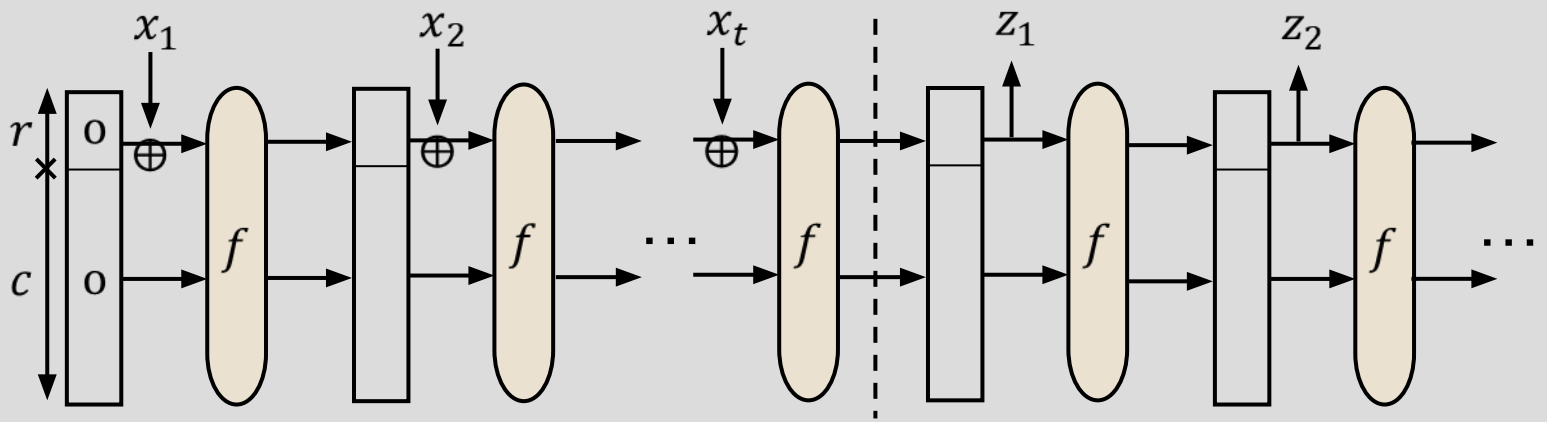
- فرض کنید f یک جایگشت شبه‌تصادفی b بیتی است.
- مقدار اولیه‌ی حالت (State) را تمام صفر در نظر می‌گیریم.
- در مرحله‌ی iام، r بیت از پیام (x_i) با r بیت از حالت xor شده و جایگشت f اعمال می‌شود.
- این کار را تکرار می‌کنیم تا پیام به صورت کامل جذب ساختار شد (Absorbing).



ساختار اسفنجی

... ادامه

- پس از آن‌که پیام به صورت کامل جذب ساختار شد، در هر مرحله r بیت از حالت استخراج شده و تابع f مجددا اعمال می‌شود (Squeezing).
- اگر طول تابع چکیده‌ساز n باشد، این کار باید $\frac{n}{r}$ بار تکرار شود.
- به r، نرخ (Rate) و به $c = b - r$ ظرفیت (Capacity) می‌گویند.



امنیت ساختار اسفنجی

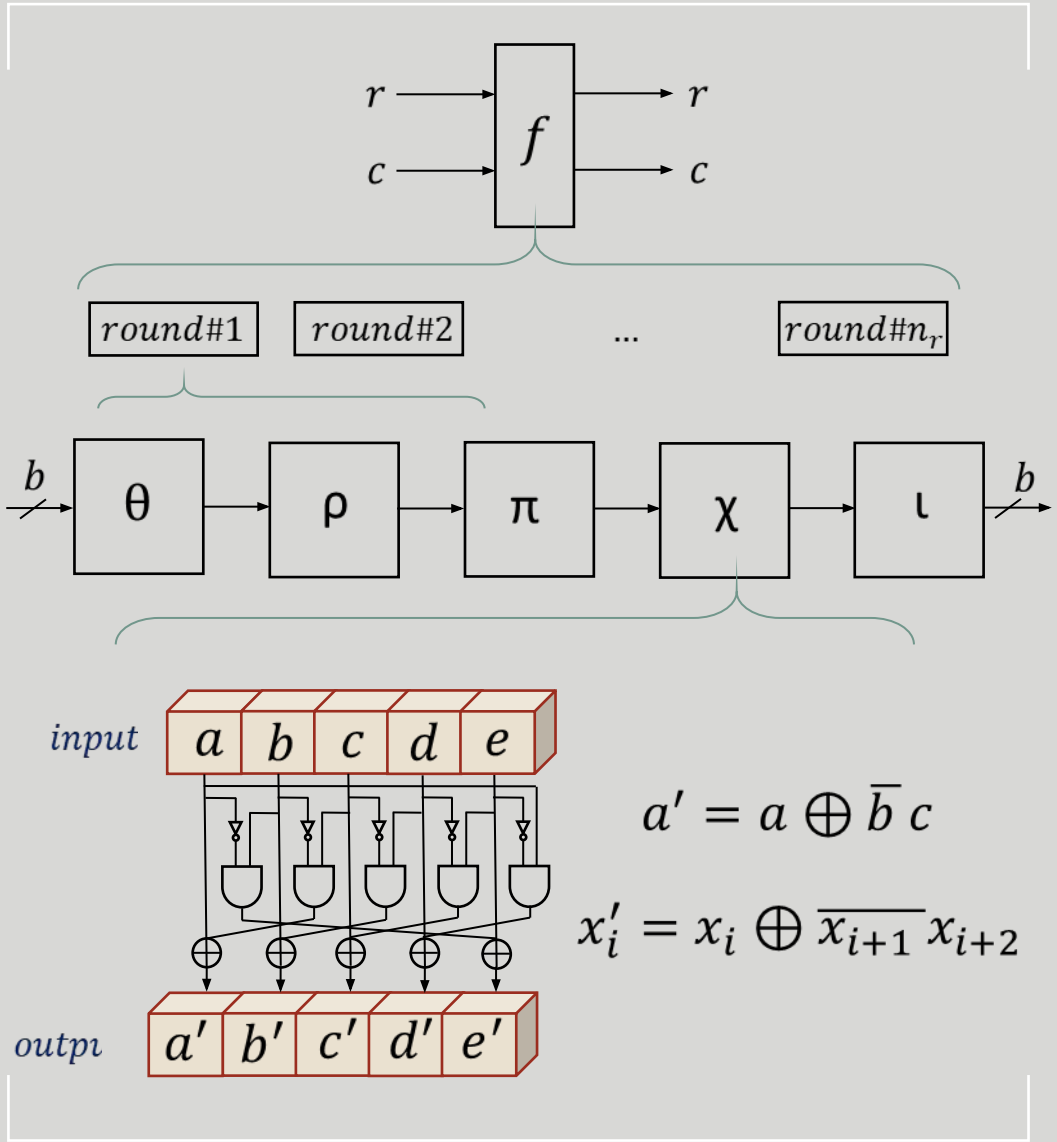
- می‌توان اثبات کرد که کران امنیتی تابع چکیده‌سازی که مبتنی بر ساختار اسفنجی است، به صورت زیر است:
  1. در مقابل پیش‌تصویر برابر $\min(2^c, 2^n)$،
  2. در مقابل پیش‌تصویر دوم $\min(2^{\frac{c}{2}}, 2^n)$.
  3. و در مقابل تصادم $\min(2^{\frac{c}{2}}, 2^{\frac{n}{2}})$.
- بنابراین ساختار اسفنجی بسیار منعطف است:

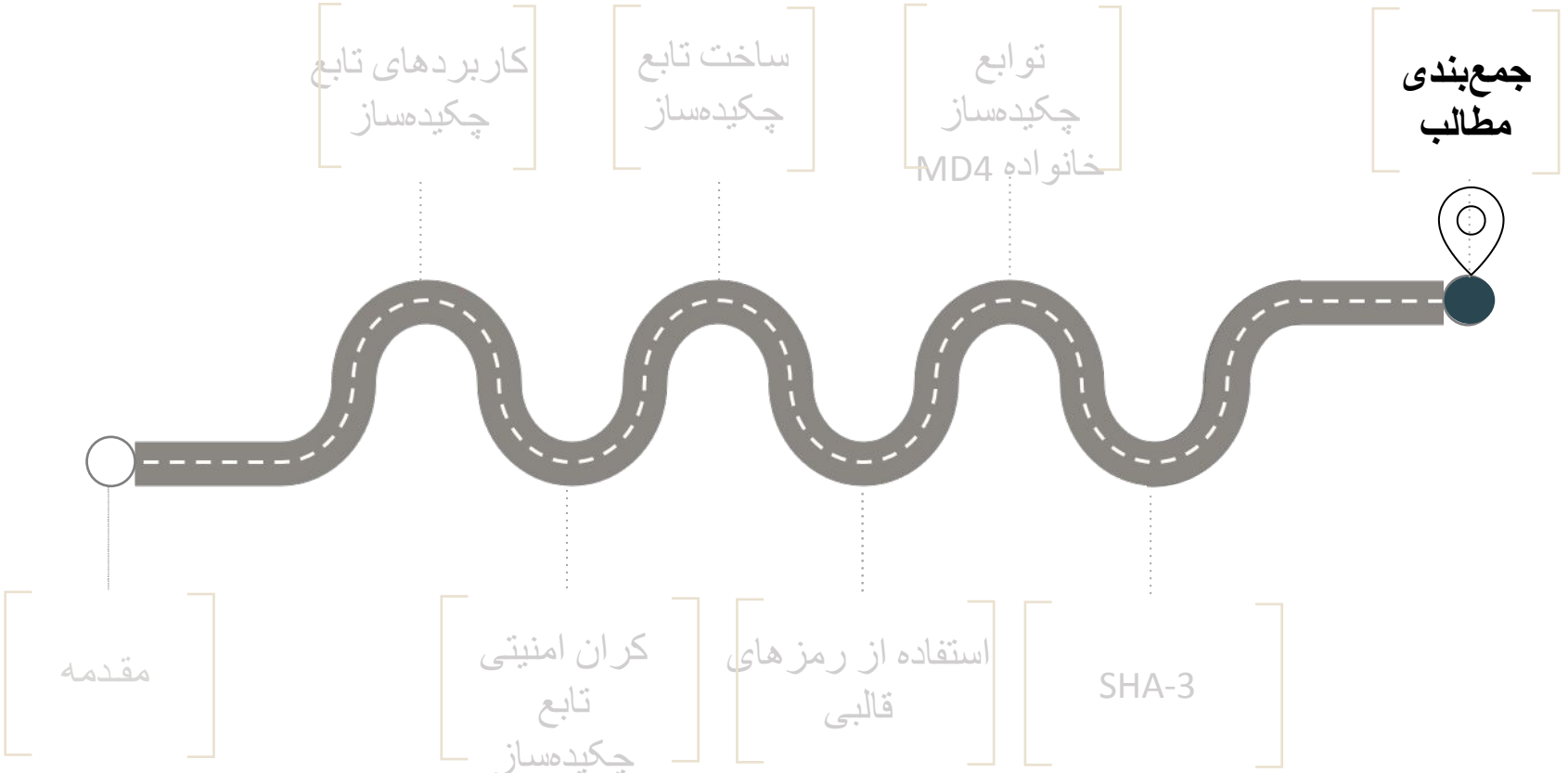
1. یک بده–بستان بین کارائی و امنیت ایجاد می‌کند: با افزایش نرخ و کاهش ظرفیت، سرعت افزایش اما امنیت کاهش می‌یابد.
2. بدون نیاز به طراحی تابعی جدید، می‌توان طول خروجی تابع چکیده‌ساز را تغییر داد.

- به همین خاطر، ساخت اولیه‌های رمزنگاری متقارن با استفاده از جایگشت شبه‌تصادفی طی سالیان اخیر مورد توجه قرار گرفته است (Permutation-Based Cryptography).

الگوریتم Keccak

- ساختار کلی الگوریتم Keccak ساختار اسفنجی است که در آن جایگشت (f) به‌کاررفته به‌صورت زیر تعریف می‌شود:
- اندازه‌ی حالت در این ساختار می‌تواند یکی از اعداد مجموعه‌ی $b = 25.2^l = \{25,50,100,200,400,800,1600\}$ باشد.
- جایگشت (f) با تکرار تابع دور ساخته می‌شود که تعداد دورها براساس رابطه‌ی $n_r = 12 + 2l$ محاسبه می‌شود.
  - به عنوان مثال برای keccak-f[200] تعداد دورها ۱۸ و برای Keccak-f[1600] تعداد دورها ۲۴ است.
- در هر دور، پنج تابع که به جز χ بقیه خطی هستند، بر روی b بیت حالت اعمال می‌شوند.
- تابع غیرخطی χ ورودی‌ها را به صورت ۵ بیت، ۵ بیت دریافت و پردازش می‌کند.




جمع‌بندی مطالب
توابع چکیده‌ساز خانواده MD4
ساخت تابع چکیده‌ساز
کاربردهای تابع چکیده‌ساز
مقدمه
کران امنیتی تابع چکیده‌ساز
استفاده از رمزهای قالبی
SHA-3

جمع‌بندی مطالب

- ویژگی‌های یک تابع چکیده‌ساز امن شامل مقاومت در برابر پیش‌تصویرها (اول و دوم) و عدم داشتن برخورد است.
- تابع چکیده‌ساز امن، کاربردهای متعددی در تامین امنیت دارد؛ احراز جامعیت پیام و ... از این جمله هستند.
- به علت نبود هیچ پارامتر مخفی، طراحی یک تابع چکیده‌ساز امن کار مشکلی است.
- پس از مشخص شدن آسیب‌پذیری MD5، MD4 و SHA-1 تابع چکیده‌ساز SHA-3 استاندارد شد.
- SHA-2 تاکنون امن باقی مانده است اما با این حال SHA-1 علی‌رغم برخی ضعف‌های شناخته شده در مقابل حمله‌ی تصادم، همچنان پرکاربردترین الگوریتم چکیده‌ساز است.