



یاد‌الامن والامان

امنیت داده و شبکه

کدهای احراز صحت پیام و توابع چکیده‌ساز

مرتضی امینی - سیدمهدی خرازی

نیم‌سال دوم ۱۴۰۴-۱۴۰۳



فهرست مطالب

□ مفاهیم اولیه

□ کدهای احراز صحت پیام

□ اصول توابع چکیده‌ساز

□ توابع چکیده‌ساز مهم

□ الگوریتم HMAC



فهرست مطالب

□ مفاهیم اولیه

□ کدهای احراز صحت پیام

□ اصول توابع چکیدهساز

□ توابع چکیدهساز مهم

□ الگوریتم HMAC



احراز صحت پیام چیست؟

اطمینان از:

۱- صحت محتوای پیام؛ یعنی پیام دریافتی دستکاری نشده است:

بدون تغییر

بدون درج

بدون حذف

۲- پیام از جانب فرستنده ادعا شده ارسال شده است.



احراز صحت پیام

- در بسیاری از کاربردها، مثلاً تراکنش‌های بانکی، حفظ محرمانگی محتوای ارتباطات اهمیت زیادی ندارد، ولی اینکه محتوای آنها قابل اعتماد باشند از اهمیت بسیار بالاتری برخوردار است.

- نیاز به دو عنصر کارکردی داریم:
 - **عنصر اول:** یک تابع برای تولید عامل احرازکننده
 - **عنصر دوم:** یک پروتکل که با استفاده از تابع فوق فوق اصالت پیام را احراز کند.



فهرست مطالب

□ مفاهیم اولیه

□ کدهای احراز صحت پیام

□ اصول توابع چکیدهساز

□ توابع چکیدهساز مهم

□ الگوریتم HMAC

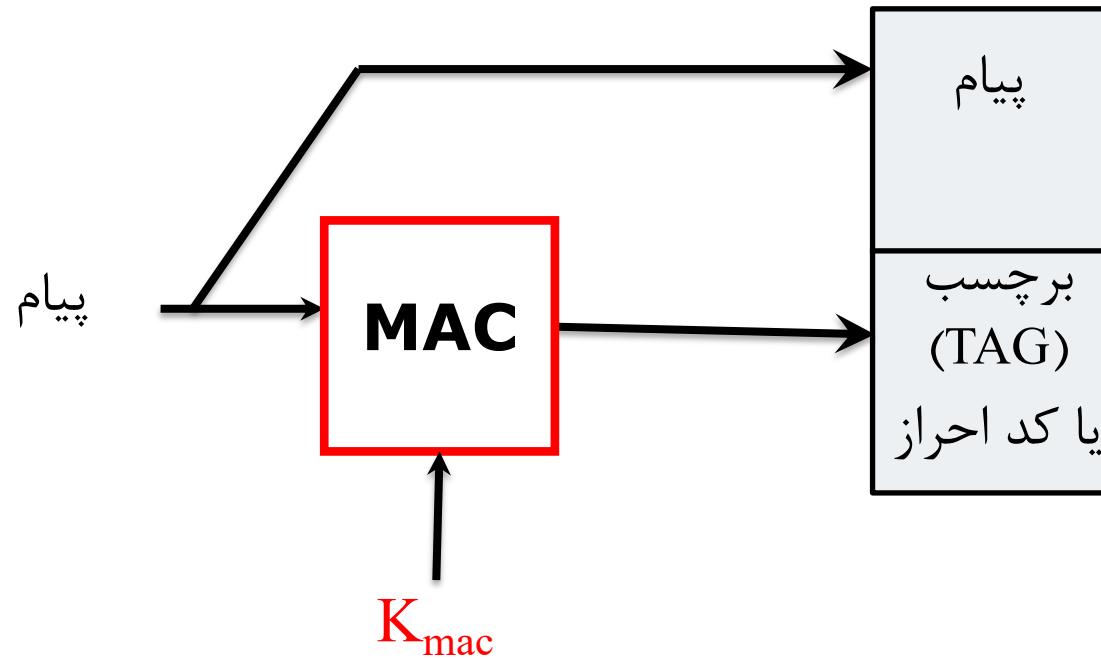


کد های احراز صحت پیام

- تولید یک برچسب با طول ثابت:
 - وابسته به **پیام** و **کلید**
 - لزوماً برگشت پذیر نیست (بر خلاف توابع رمزنگاری)
 - نیازمند اشتراك یک کلید مخفی بین طرفین
 - آنرا به اختصار **MAC** مینامند. نام دیگر "Cryptographic Checksum"
 - این برچسب را به پیام اضافه می‌کنند.
 - گیرنده برچسب پیام را محاسبه نموده و با برچسب ارسالی مقایسه می‌کند.
 - از صحت پیام و هویت فرستنده اطمینان حاصل می‌شود.



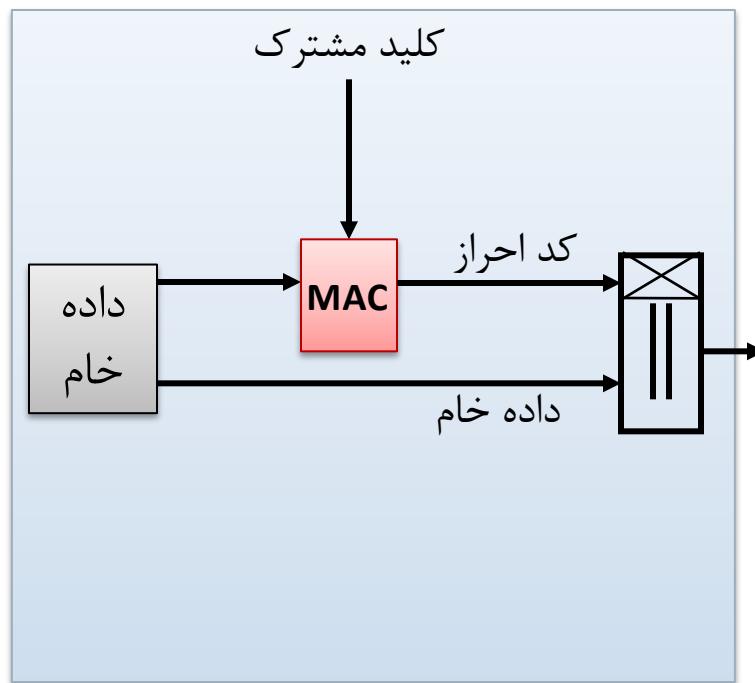
کد های احراز صحت پیام



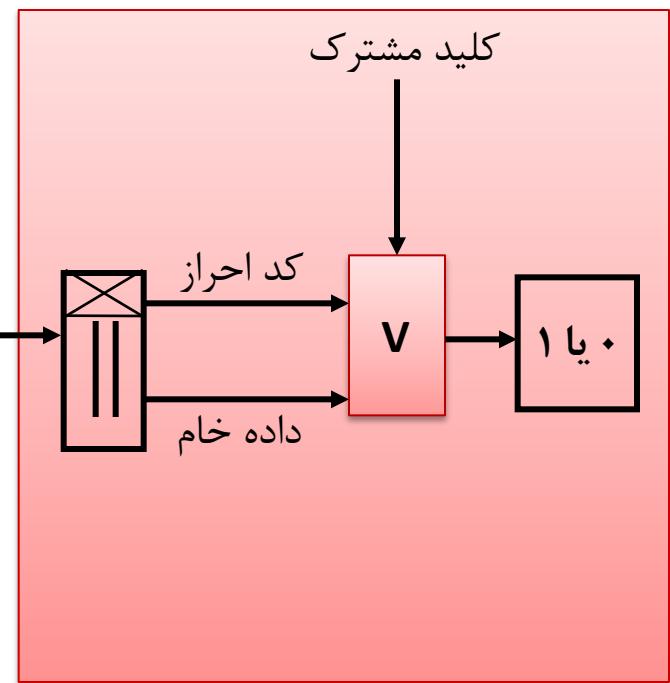


نحوه عملکرد کدهای احراز صحت پیام

حسن



علی



V: Verification



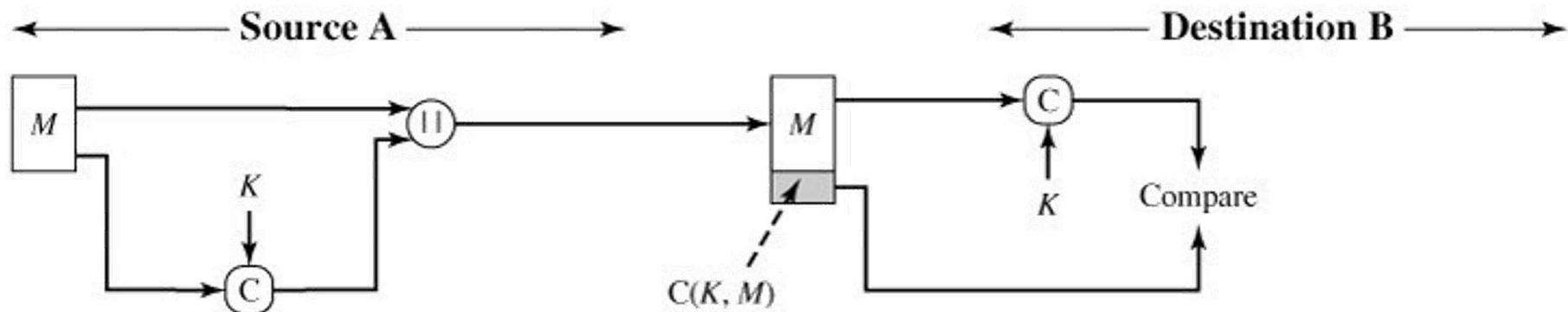
فرق MAC و رمزگذاری

- MAC نیازی ندارد که حتماً برگشت پذیر باشد، در صورتی که الگوریتم رمزگذاری باید برگشتپذیر باشد.
- MAC تابع چند به یک است.
- اندازه خروجی MAC برابر n بیت، تعداد MAC‌های ممکن = 2^n
- اندازه کلید MAC برابر k بیت، تعداد نگاشتهای ممکن به MAC‌ها = 2^k
- با توجه به خصوصیات ریاضی MAC، آسیب‌پذیری‌های احتمالی برای شکست آن کمتر است.



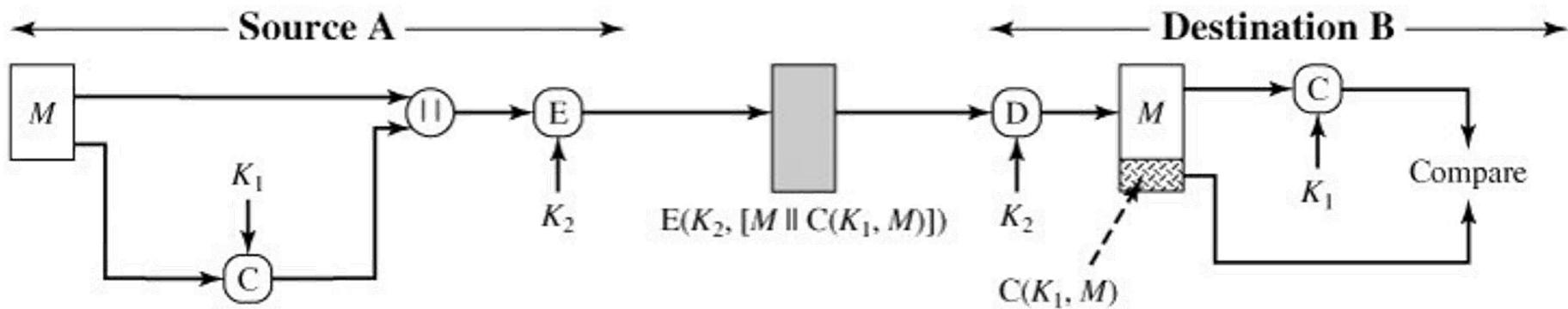
نحوه کاربرد کدهای احراز صحت پیام

احراز صحت پیام



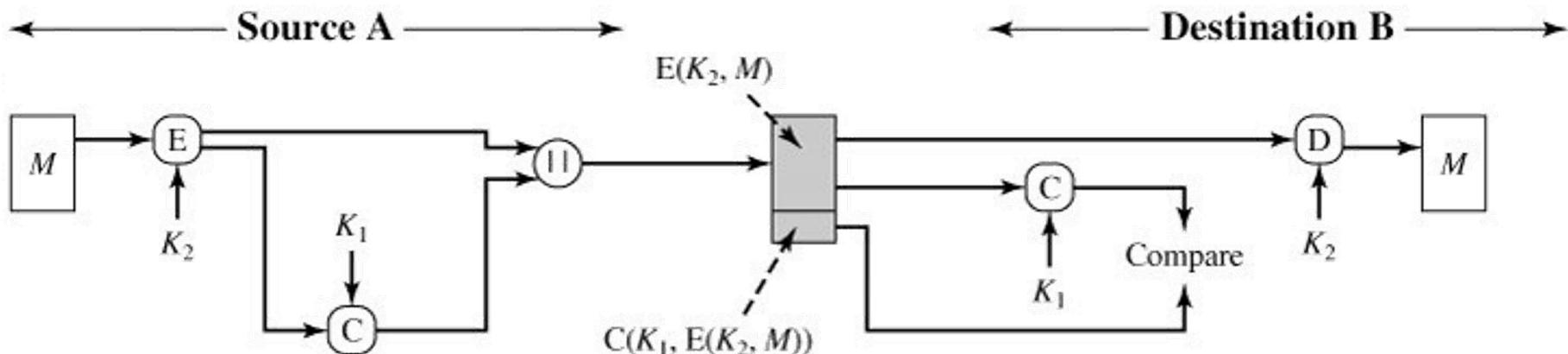
نحوه کاربرد کدهای احراز صحت پیام

احراز صحت پیام همراه با محرمانگی؛ احراز صحت پیام آشکار



نحوه کاربرد کدهای احراز صحت پیام

احراز صحت پیام همراه با محرمانگی؛ احراز صحت پیام رمز





امنیت MAC

□ حمله آزمون جامع به کلید MAC

- با داشتن یک متن و MAC آن، به صورت برون خط انجام می‌پذیرد.
- اگر طول کلید k بیت باشد، 2^k کلید ممکن باید بررسی شود.
- با یافتن یک کلید، باید آن را با زوجهای دیگری چک کرد، چون ممکن است چند کلید مختلف، یک متن را به چکیده یکسان نگاشت کنند.

□ حمله آزمون جامع برای کشف تصادم

- با داشتن یک MAC، به دنبال پیامی می‌گردیم که همان MAC را حاصل نماید.
- اگر MAC، n بیتی باشد، به طور متوسط با 2^n پیام، به احتمال زیاد یک تصادم (collision) رخ می‌دهد.



امنیت MAC

هزینه لازم برای حمله آزمون جامع به MAC برابر است با

$$\min(2^k, 2^n)$$

ویژگیهای یک MAC مناسب:

■ با دانستن یک پیام و بر چسب آن، یافتن پیام متفاوتی با برچسب یکسان از لحاظ محاسباتی ناممکن باشد.

■ توزیع خروجی MAC باید یکنواخت باشد تا احتمال اینکه دو پیام تصادفی MAC یکسان داشته باشند، کمینه شود.

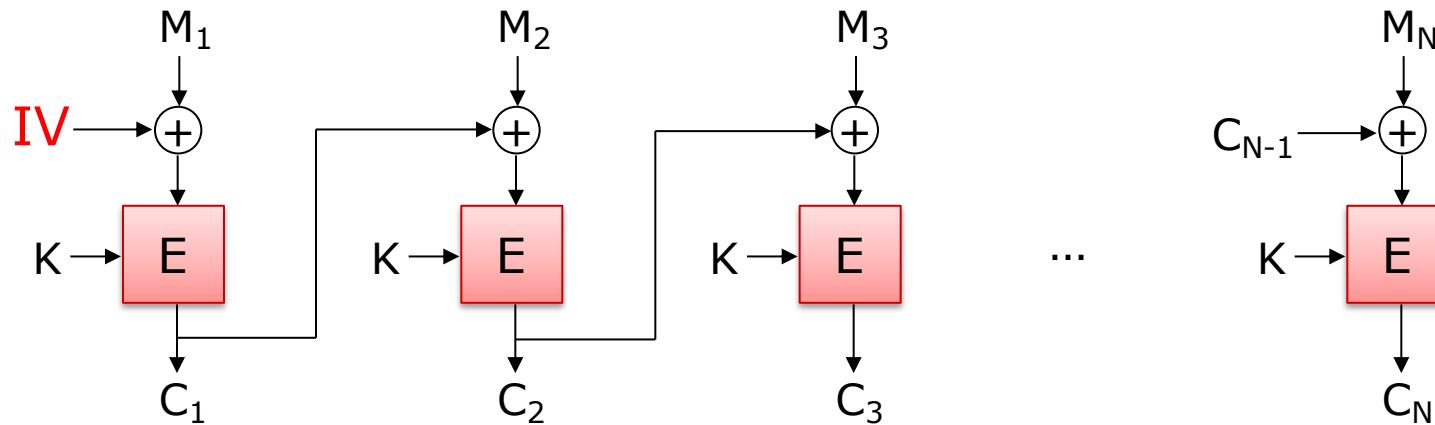
نکته: طول برچسب MAC همانند طول کلید در امنیت MAC تاثیر دارد.



ساختن MAC امن با استفاده از رمزگذاری

- با استفاده از توابع رمزگذاری امن و برخی از سبک‌های رمزنگاری می‌توان توابع MAC امن ساخت.
 - مثال: سبک‌های CFB و CBC
- **مثال:** استاندارد DAA (Data Authentication Algorithm) استاندارد NIST و ANSI X9.17
- بر اساس رمز قطعه‌ای DES و مددکاری CBC
- در ساختن MAC از این سبک‌ها باید دقت زیادی کرد.
 - جزئیات بسیار مهم‌اند.
 - در ادامه تلاش می‌کنیم تا CBC-MAC بسازیم!

۱ - تلاش ۱ – CBC-MAC



$M = (M_1, \dots, M_N)$ پیام:

$T = (IV, C_N)$ برچسب:

پیام به همراه برچسب فرستاده می‌شود.

برای احراز صحت، برچسب از نو محاسبه و با برچسب دریافتی مقایسه می‌شود.



حمله به تلاش ۱

□ مهاجم می‌تواند با انتخاب IV به دلخواه، قطعه اول پیام را تغییر دهد.

□ با داشتن پیام $M = (M_1, M_2, \dots, M_N)$ و برچسب $T = (\text{IV}, C_N)$ می‌توان پیام و برچسب جدیدی را بدون داشتن کلید جعل کرد:

$$M' = (\textcolor{red}{M'_1}, M_2, \dots, M_N)$$

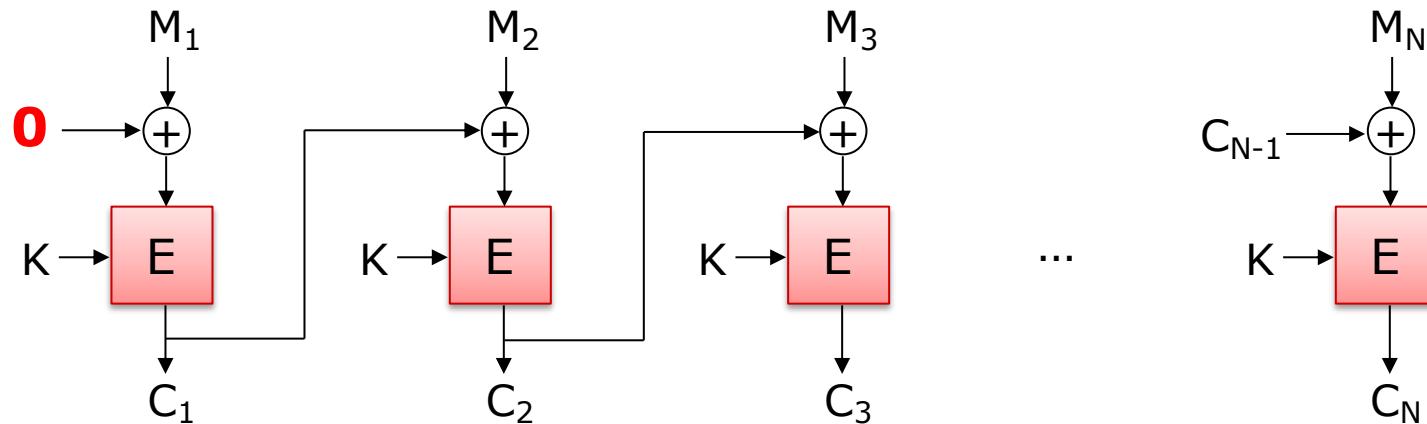
$$T' = (IV', C_N)$$

$$IV' \oplus M'_1 = IV \oplus M_1$$



۲ - تلاش CBC-MAC

- راهکار: استفاده از CBC-MAC با یک IV ثابت؛ مثلاً بردار تمام صفر.
- برچسب مساوی C_N است.





حمله به تلاش ۲ - افزایش طول (Length Extension)

با داشتن پیام تک قالبی $T = C_1$ و برچسب $M = (M_1)$ □
می‌توان پیام و برچسب جدیدی را بدون داشتن کلید جعل کرد:

$$M' = (M_1, \textcolor{green}{M}_2)$$

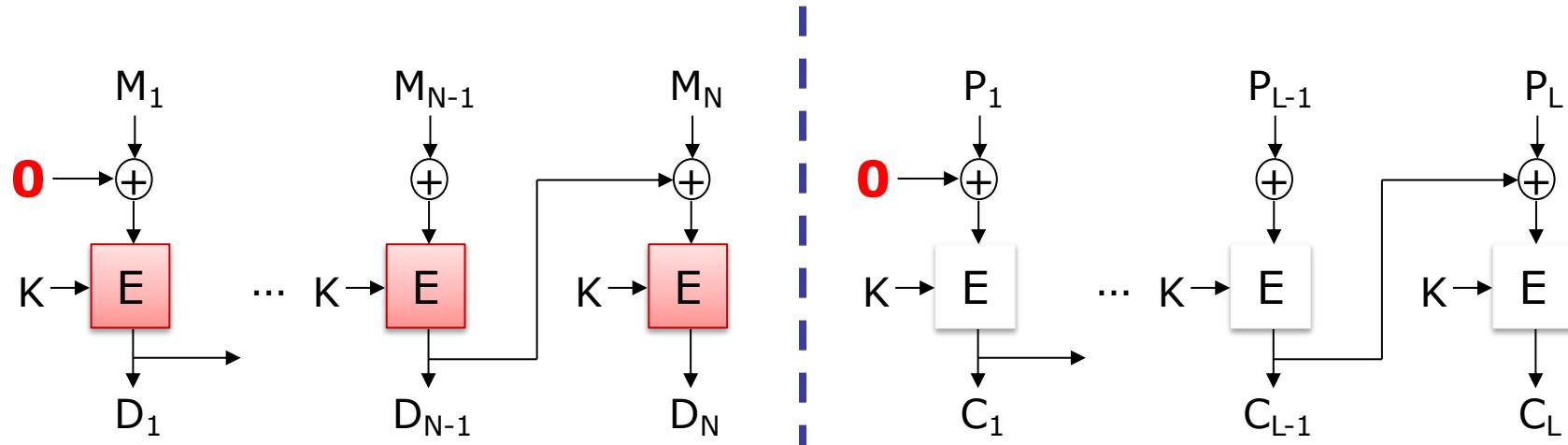
$$T' = T = C_1$$

$$\textcolor{green}{M}_2 = M_1 \oplus C_1$$

به همین ترتیب می‌توان جعل را ادامه داد و به پیامهایی با طول
بیشتر رسید. □



حمله به تلاش ۲ - برچسب جدید از دو برچسب موجود



دو پیام و برچسب روی هریک را داریم: □

پیام C_L با برچسب $P = (P_1, \dots, P_L)$ ■

پیام D_N با برچسب $M = (M_1, \dots, M_N)$ ■

$$M' = (M_1, \dots, M_N, D_N \oplus P_1, P_2, \dots, P_L)$$

$$T' = T = C_L$$



راهکارها

□ راهکار ۱: همه پیامهای سیستم، طول N داشته باشند.

- جلوگیری از حمله افزایش طول
- مناسب برای بسیاری از پروتکلها

□ راهکار ۲: همیشه طول پیام را به عنوان قطعه اول به تابع

CBC-MAC می‌دهیم.

□ راهکار ۳: قطعه آخر (C_N) را یک مرتبه مجدداً رمز می‌کنیم.

□ اثبات شده است که همه راهکارهای فوق امن هستند.



فهرست مطالب

- مفاهیم اولیه
- گدهای احراز صحت پیام
- اصول توابع چکیده‌ساز
- توابع چکیده‌ساز مهم
- الگوریتم HMAC



توابع چکیده‌ساز (Hash Functions)

- توابع چکیده‌ساز کاربردهای مختلفی در امنیت دارند که مهم‌ترین آن در احراز اصالت پیام و امضای دیجیتال است.
- خصوصیات:
 - تابع یک‌طرفه
 - طول ورودی متغیر
 - طول خروجی ثابت (نگاشت از فضای بزرگ‌تر به فضای کوچک‌تر)
 - در حالت کلی، کلیدی در کار نیست! بر خلاف MAC و رمزنگاری



امنیت توابع چکیده‌ساز – ایده کلی

- نگاشت پیام‌های طولانی به رشته‌های کوتاه به گونه‌ای که:
- یافتن پیام‌های متفاوتی که به یک رشته یکسان نگاشته شوند دشوار باشد.
- به این رشته، **عصاره** یا **چکیده پیام** (Message Digest) می‌گوییم.



نیازمندیهای توابع چکیده‌ساز

- توابع چکیده‌ساز باید یک طرفه (One-Way) باشند.
- برای یک h داده شده، باید یافتن x به گونه‌ای که $(x) = H(x)$ از لحاظ محاسباتی ناممکن باشد.
- مقاومت در برابر تصادم ضعیف (Weak Collision)
 - برای یک x داده شده، باید یافتن y به گونه‌ای که $(y) = H(x)$ باشد، از لحاظ محاسباتی ناممکن باشد.
- مقاومت در برابر تصادم قوی (Strong Collision)
 - یافتن x و y به گونه‌ای که $(x) = H(y) = H(y)$ باشد، از لحاظ محاسباتی ناممکن باشد.



مقایسه تصادم قوی و ضعیف

□ ممکن است ساختار تابع H طوری باشد که:

■ بتوان تعداد محدودی x و y یافت به گونه‌ای که مقادیر تابع، تصادم پیدا کنند (تصادم قوی).

■ ولی برای یک x داده شده همواره نتوان یک y پیدا کرد بطوریکه $H(y) = H(x)$ (تصادم ضعیف).

◀ ارضاشدن شرط عدم وجود تصادم قوی برای یک تابع دشوارتر از ارضاشدن شرط عدم وجود تصادم ضعیف است.

◀ توابعی که در برابر تصادم قوی مقاومت کنند امنیت بالاتری دارند.



امنیت توابع چکیده‌ساز

□ توابع چکیده‌ساز باید یک طرفه باشند.

- پیچیدگی جستجوی کامل (آزمون جامع) برای یافتن یک **پیش‌نگاره** (یعنی یک مقدار x که $H(x)=h$ باشد) n^2 است، که n طول خروجی تابع است.

□ مقاومت در برابر تصادم (ضعیف)

با کمک حمله
روز تولد

- پیچیدگی جستجوی کامل (آزمون جامع) n^2 است.

□ مقاومت در برابر تصادم (قوی)

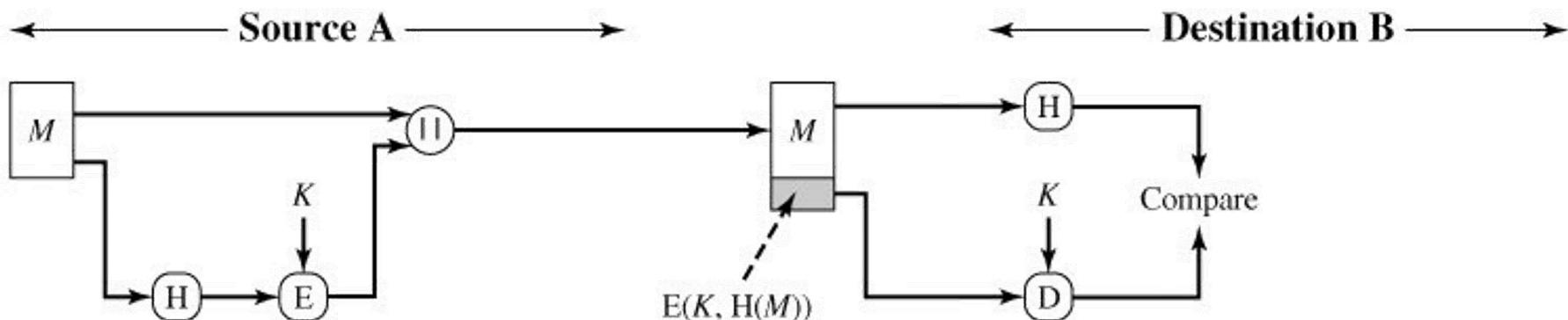
- پیچیدگی جستجوی کامل (آزمون جامع) $n^{1/2}$ است.

□ دقت کنید آزمون جامع بیانگر حداکثر امنیت ممکن برای تابع است، زیرا ممکن است به دلیل ضعف طراحی، حملات موثرتری نیز امکان پذیر باشد.

کاربرد توابع چکیده‌ساز

احراز صحت پیام در ترکیب با رمز متقاضی

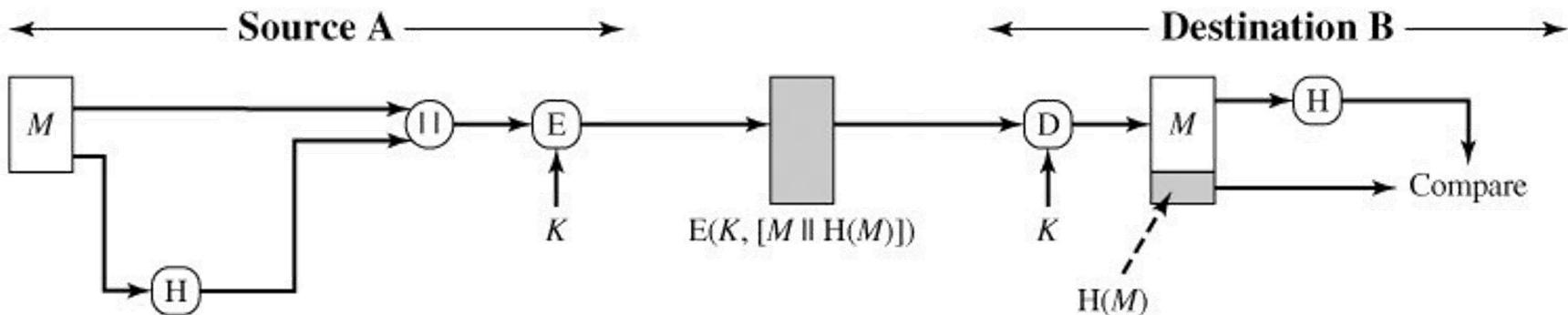
- صرفاً رمزگذاری چکیده پیام
- این ترکیب در واقع یک کد احراز صحت پیام (MAC) را می‌سازد.





کاربرد توابع چکیده‌ساز

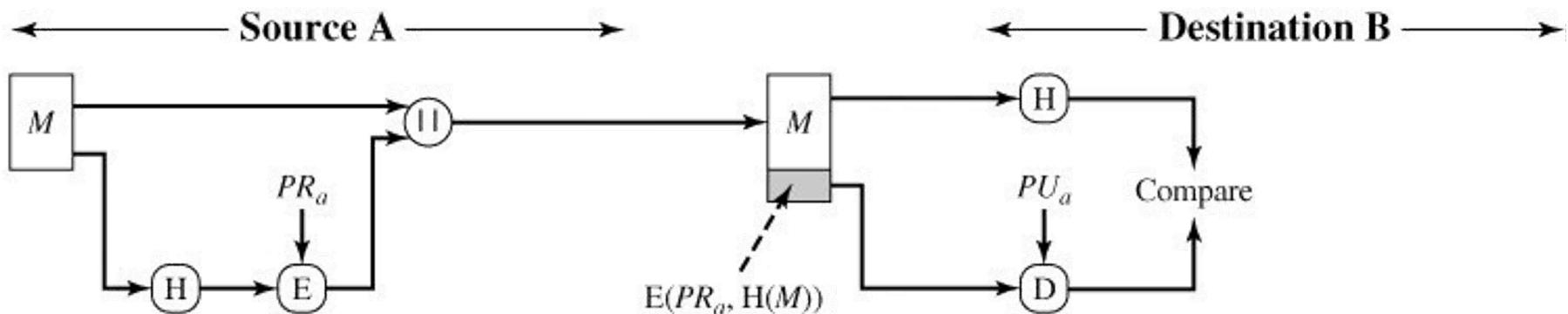
احراز صحت پیام و محربانگی در ترکیب با رمز متقارن



کاربرد توابع چکیده‌ساز

احراز صحت پیام در ترکیب با رمز کلید عمومی

- صرفاً رمزگذاری چکیده پیام
- این ترکیب در واقع یک امضای دیجیتال را می‌سازد.

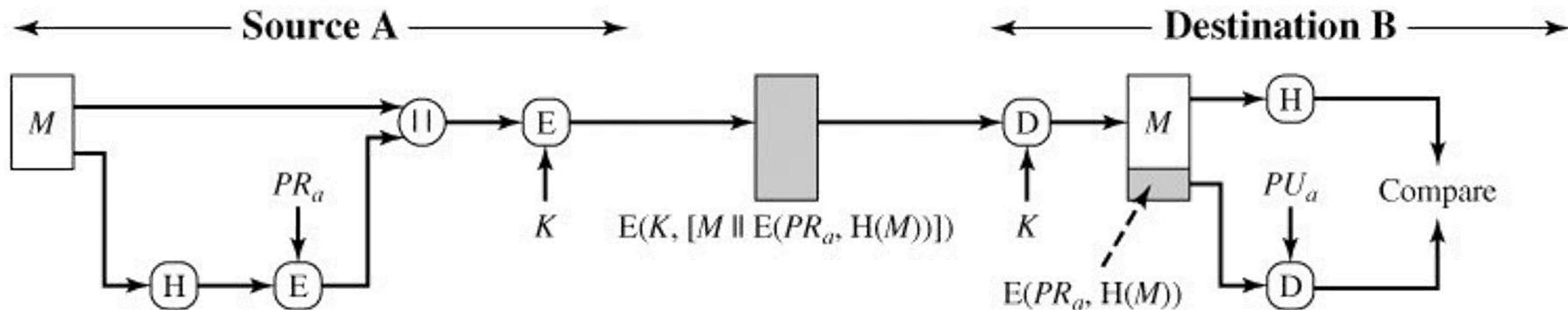


کاربرد توابع چکیده‌ساز

احراز صحت پیام و محرمانگی در ترکیب با رمز متقارن و نامتقارن

- امضای دیجیتال با رمز نامتقارن برای حفظ صحت

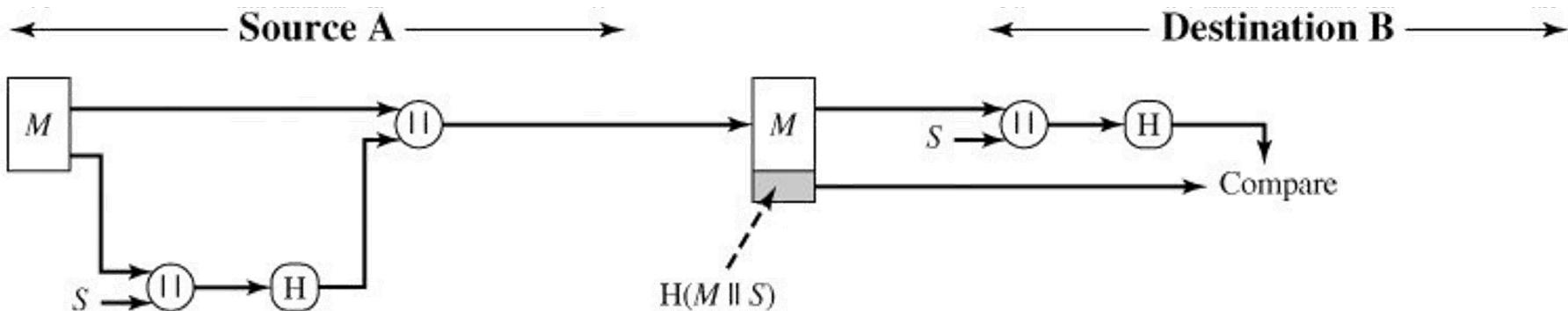
- رمز متقارن برای حفظ محرمانگی



کاربرد توابع چکیده‌ساز

احراز صحت پیام بدون رمزگذاری

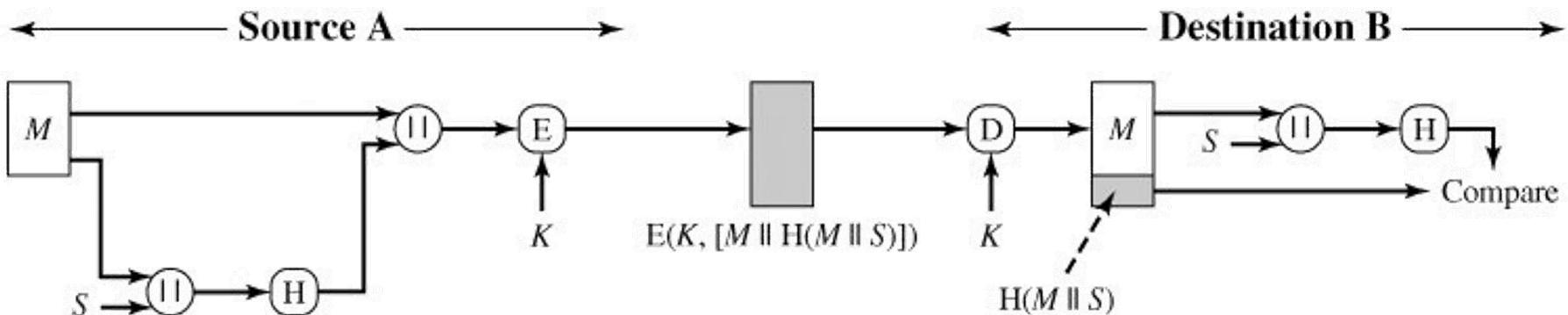
- طرفین راز S را مخفیانه به اشتراک می‌گذارند.
- بدون استفاده از رمزگذاری
- کاربرد عملی زیاد ولیکن ممکن است آسیب‌پذیر در برابر حمله افزایش طول باشد.
(توضیح در اسلایدهای بعدی)



کاربرد توابع چکیده‌ساز

احراز صحت پیام بدون رمزگذاری و محرمانگی با رمز متقارن

- رمزگذاری صرفاً برای محرمانگی





پارادوکس روز تولد



- در میان ۲۳ نفر، احتمال یافتن دو نفر که در یک روز از سال متولد شده باشند، بیش از ۵۰٪ است.
- این پارادوکس به دو شکل عمومی برای یک **تابع دلخواه توسعه** یافت.
- اندازه حداقل **یک** مجموعه برای یافتن یک زوج در آن با خروجی یکسان با احتمال بیش از ۵/۰
- اندازه حداقل **دو** مجموعه برای یافتن یک تصادم بین اعضای آنها با احتمال بیش از ۵/۰



پارادوکس روز تولد

□ مبانی ریاضی

- تابع H با 2^n خروجی ممکن را در نظر بگیرید (خروجی n بیتی).
- به H , k ورودی تصادفی اعمال کنیم و خروجی را مجموعه X در نظر می‌گیریم.
- به همین ترتیب مجموعه Y را تشکیل می‌دهیم.
- اگر k بزرگتر از $2^{n/2}$ باشد، احتمال حداقل یک تصادم در بین اعضای دو مجموعه X و Y بیش از 50% است.



حمله روز تولد

- ممکن است تصور کنید یک Hash ۶۴ بیتی امن است اما با حمله روز تولد امنیت از بین می‌رود:
- مهاجم $2^{n/2}$ پیام **معتبر** که اساساً هم معنا هستند تولید می‌کند. n طول خروجی Hash است.
- مهاجم همین تعداد از گونه‌های هم معنا از پیام **بدخواهانه دلخواه** خود را تولید می‌کند.
- دو دسته پیام مقایسه می‌شوند تا زوجی یافت شود که چکیده یکسان داشته باشند.
- از کاربر می‌خواهیم تا پیام **معتبر** را امضا نماید، و سپس پیام **بدخواهانه دلخواه** مهاجم را جایگزین می‌کنیم.



مثالی از حمله (پیام معتبر)

Dear Dean Smith,

This [letter | message] is to give my [honest | frank] opinion of Prof Tom Wilson, who is [a candidate | up] for tenure [now | this year]. I have [known | worked with] Prof Wilson for [about | almost] six years. He is an [outstanding | excellent] researcher of great [talent | ability] known [worldwide | internationally] for his [brilliant | creative] insights into [many | a wide variety of] [difficult | challenging] problems.



مثالی از حمله (پیام جعلی مهاجم)

Dear Dean Smith,

This [letter | message] is to give my [honest | frank] opinion of Prof Tom Wilson, who is [a candidate | up] for tenure [now | this year]. I have [known | worked with] Prof Wilson for [about | almost] six years. He is an [poor | weak] researcher not well known in his [field | area]. His research [hardly ever | rarely] shows [insight in | understanding of] the [key | major] problems of [the | our] day.



ساختار مرکل-دَمَگَارد برای توابع چکیده‌ساز

- مورد استفاده در بسیاری از توابع چکیده‌ساز
- اعمال مکرر یک تابع فشرده‌ساز به یک رشته با طول ثابت
- اگر تابع فشرده‌ساز مقاوم در برابر تصادم باشد، تابع چکیده‌ساز نیز همین‌گونه خواهد بود.
- توابع معروفی مانند MD5 و SHA-1 از همین ایده استفاده می‌کنند.

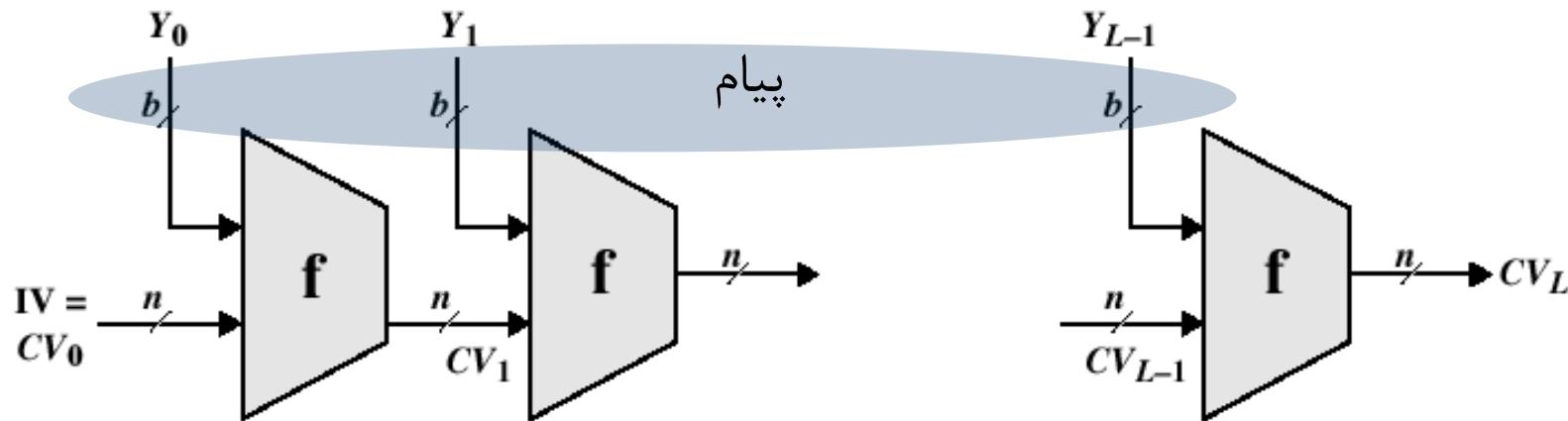


Ralph Merkle
(1952 -)



Ivan Bjerre
Damgård (1956 -)

ساختار مرکل-دمگارد برای توابع چکیده‌ساز



IV = Initial value
 CV = chaining variable
 Y_i = i th input block
 f = compression algorithm
 L = number of input blocks
 n = length of hash code
 b = length of input block

- پیام به قطعات Y_i تقسیم شده است.
- IV یک رشته ثابت می‌باشد.

$$\begin{aligned}
 CV_0 &= IV \\
 CV_i &= f(CV_{i-1}, Y_{i-1}) \\
 \text{Hash} &= CV_L
 \end{aligned}$$



فهرست مطالب

□ مفاهیم اولیه

□ گدهای احراز صحت پیام

□ اصول توابع چکیدهساز

□ توابع چکیدهساز مهم

□ الگوریتم HMAC



توابع چکیده‌ساز مهمنما: MD5

MD5: Message Digest 5 □

- طراحی 1992 توسط ران ریوست، یکی از سه طراح RSA □ استفاده گسترده در گذشته، اما از کاربرد آن کاسته شده است.



- ویژگیها:
 - پیام به قطعات ۵۱۲ بیتی تقسیم می‌شود.
 - خروجی ۱۲۸ بیتی



امنیت MD5

- مقاومت در برابر تصادم (قوى) تحت حمله آزمون جامع: ۲۶۴
- امروزه امن محسوب نمی‌شود.

- حملات کاراتری به این الگوریتم یافت شده‌اند:
- سال ۱۹۹۲: حمله تفاضلی به یک دور الگوریتم Berson
- سال ۱۹۹۶: تصادم در تابع فشرده‌ساز Dobbertin



توابع چکیده‌ساز مهه: SHA-1

SHA-1: Secure Hash Algorithm – 1 □

استاندارد NIST، ۱۹۹۵

طول ورودی کوچکتر از 2^{64} بیت

طول خروجی ۱۶۰ بیت

استفاده شده در استاندارد امضای دیجیتال DSS

امنیت: □

مقاومت در برابر تصادم (قوی) تحت حمله روز تولد: 2^{80}

در سال ۲۰۰۵ توانستند با 2^{69} عمل یک تصادم در آن بیابند.

در آمریکا از سال ۲۰۱۰ الزام شد که با گونه‌های امن‌تر آن یعنی خانواده SHA-2 جایگزین شود.



توابع چکیده‌ساز مهه: SHA-2

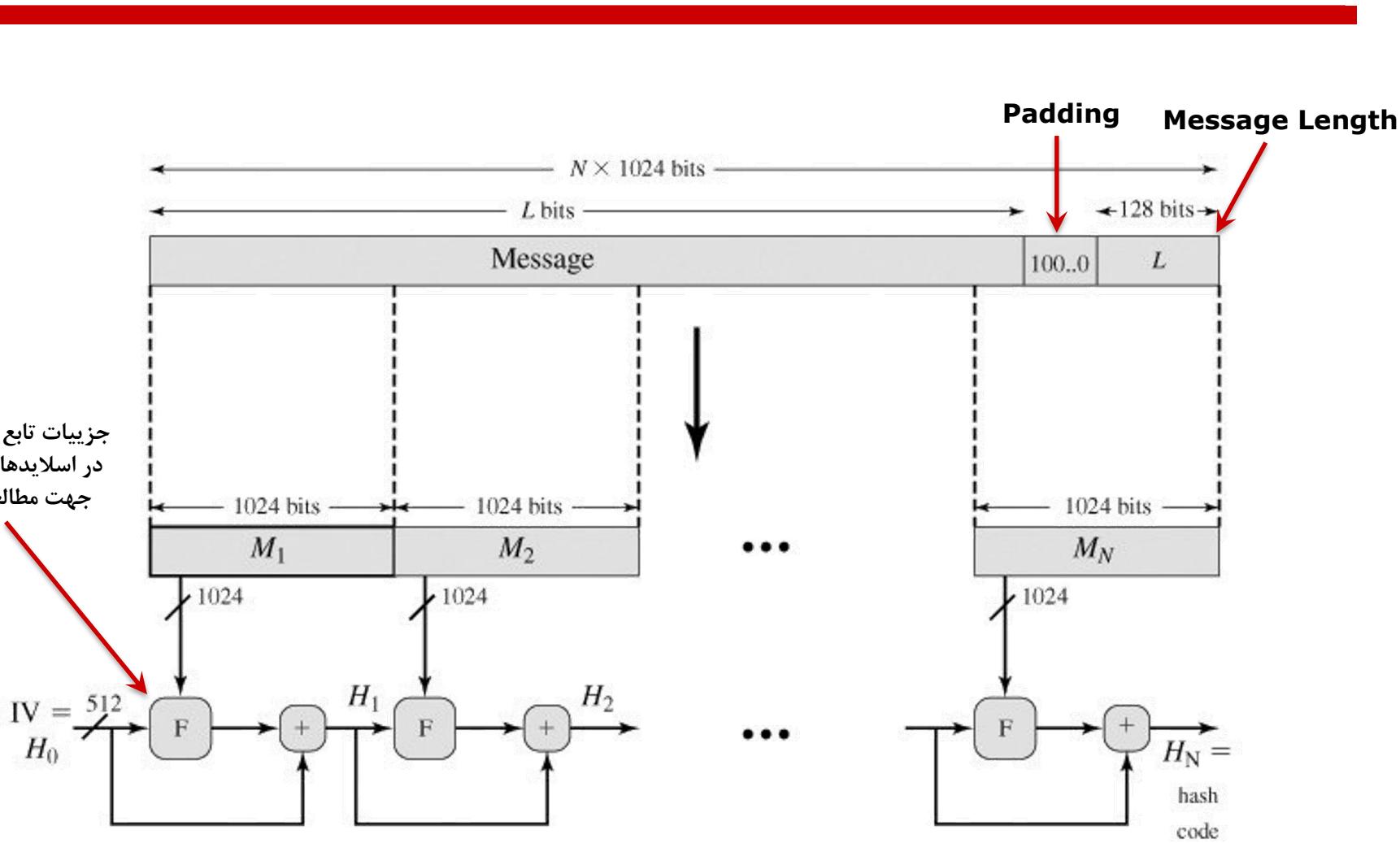
نسخه‌های زیر نیز علاوه بر SHA-1 استاندارد شده‌اند:

- معروف به **خانواده SHA-2** هستند.
- از لحاظ ساختار و جزئیات مشابه SHA-1 هستند.
- چهار تابع آخر در ماشین‌های ۶۴ بیتی از دو تابع دیگر خانواده SHA-2 سریعتر هستند (هر چند طول خروجی تابع اولیه بیشتر است).

Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256



الگوریتم SHA-512





حمله افزایش طول

SHA-1، MD5 و تمامی توابع خانواده SHA-2 در برابر حملات

افزایش طول آسیب‌پذیر هستند.

- اگر بخواهیم پیام m را به صورت $H(K \parallel m)$ احراز صحت کنیم، مهاجم با دانستن m ، برای مقدار دلخواه m' می‌تواند به سادگی مقدار $H(K \parallel m \parallel pad \parallel L \parallel m')$ را بدست آورد.
- در این حالت، مهاجم می‌تواند m را با $m \parallel pad \parallel L \parallel m'$ جایگزین نماید.



حمله افزایش طول

□ برای حل این مشکل، می‌توان:

- طول پیام را به عنوان قطعه نخست به ساختار تابع چکیده‌ساز داد.
- برای قطعه آخر، از یک تابع فشرده‌ساز متفاوت بهره گرفت.
- کلید را بعد از پیام به صورت $k \parallel m$ قرار داد و سپس H را اعمال کرد.



تابع چکیده‌ساز SHA-3

- NIST در سال ۲۰۰۷ مسابقه‌ای را برای انتخاب تابع چکیده‌ساز جدید و معرفی آن به عنوان استاندارد **SHA-3** آغاز کرد.
- از شرایط SHA-3 آن بود که حمله افزایش طول به آن وارد نباشد.
- ساختارهای جدید (غیر مرکل-دمگارد) بیشتر مورد استقبال قرار گرفت.
- SHA-3 در حال حاضر به عنوان جایگزین SHA-2 مطرح نیست و هر دو الگوریتم به عنوان الگوریتم‌های استاندارد قابل استفاده هستند.



استاندارد SHA-3

□ در سال ۲۰۱۲ تابع چکیده‌ساز Keccak به عنوان برنده و تابع ۳ تعیین گردید.

■ تلفظ رسمی: Catch-Ack

■ ساختار: توابع اسفنجی (Sponge)

■ طراحان: Van Rijndael، Peeters، Daemen، Bertoni و Assche

□ استاندارد SHA-3 در آگوست ۲۰۱۵ منتشر شد.

□ طول ورودی: دلخواه

□ طول خروجی: ۱۲۸، ۲۵۶، ۳۸۴، ۵۱۲، ۷۶۸ و دلخواه.



فهرست مطالب

□ مفاهیم اولیه

□ کدهای احراز صحت پیام

□ اصول توابع چکیدهساز

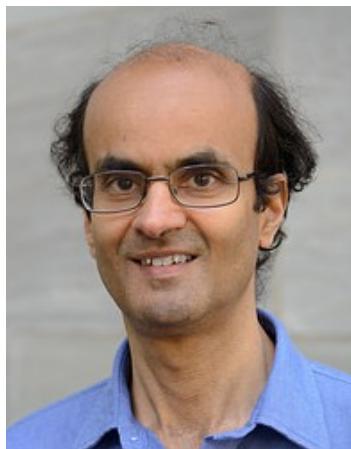
□ توابع چکیدهساز مهم

□ الگوریتم HMAC



کد احراز اصالت HMAC

□ ابداع توسط بلاری، کانتی و کرفچیک در سال ۱۹۹۶.



Mihir Bellare



Ran
Canetti



Hugo
Krawczyk



کد احراز اصالت HMAC

- **HMAC** یک الگوریتم احراز صحت پیام است.
- **HMAC** اساساً روشی برای ترکیب کردن کلید مخفی با الگوریتم‌های چکیده‌ساز فعلی است.
- مانند RIPEMD-160، Whirlpool، SHA-2، SHA-1، MD5 و
- حمله افزایش طول به **HMAC** وارد نیست.
- برای تولید چکیده پیام، از توابع چکیده‌ساز استفاده شده است.
- در مقابل استفاده از رمزهای قطعه‌ای
- بدلیل مزایای عملی توابع چکیده‌ساز



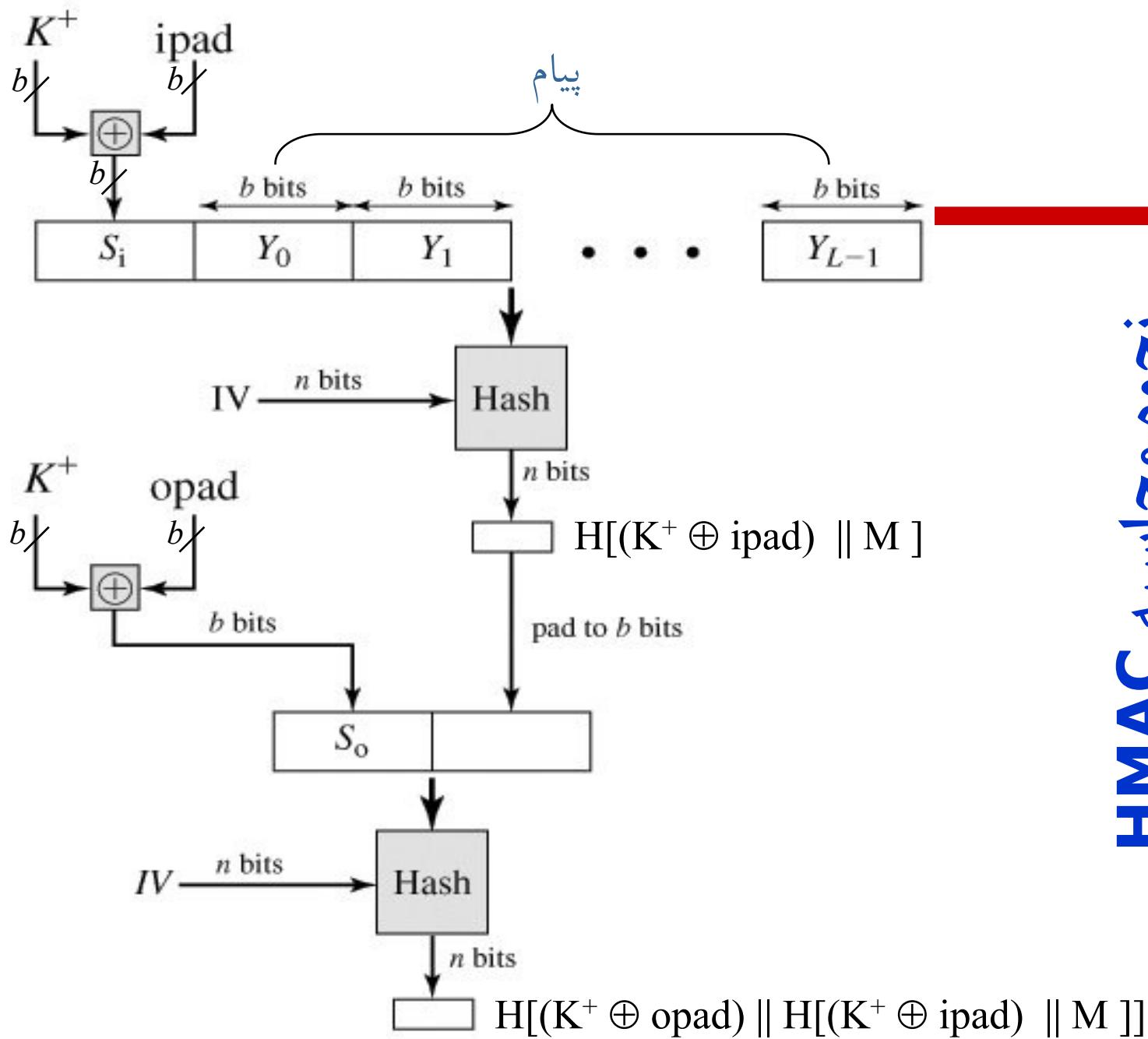
الگوریتم HMAC

- H : تابع چکیده‌ساز به کار گرفته شده (با خروجی n بیتی)
- M : پیام ورودی (با قطعات b بیتی)
- K : کلید مخفی (طول پیشنهادی بیشتر از n ، در صورتیکه طول بیشتر از b بیت باشد، کاهش به n بیت با استفاده از توابع چکیده‌ساز)
- K^+ : کلید مخفی که یک دنباله صفر به سمت چپ آن اضافه شده است (تا به طول b برسد)
- $b/8$: رشته b بیتی حاصل از تکرار رشته $0\ 1\ 1\ 0\ 1\ 1\ 0\ 0$ به تعداد $ipad$
- $b/8$: رشته b بیتی حاصل از تکرار رشته $1\ 0\ 1\ 1\ 0\ 1\ 0\ 0$ به تعداد $opad$

$$HMAC(K,M) = H[(K^+ \oplus opad) \parallel H[(K^+ \oplus ipad) \parallel M]]$$



نحوه محاسبه HMAC





امنیت HMAC

- ارتباط دقیق بین امنیت HMAC با امنیت تابع در همساز اثبات شده است.
- حمله به HMAC نیاز دارد به
 - حمله آزمون جامع بر روی کلید (میزان مقاومت بسته به طول کلید)
 - حمله روز تولد که با توجه به نداشتن کلید نیازمند مشاهده تعداد زیادی پیام و MAC آنهاست که از کلید یکسانی در آنها استفاده شده است.
- مقاومت HMAC در برابر حمله روز تولد از تابع چکیده‌ساز به کار گرفته شده، بیشتر است.
- لذا استفاده از MD5 در هنگام نیاز به سرعت بیشتر مجاز است.



پایان

پست الکترونیکی

amini@sharif.edu

kharrazi@sharif.edu

یادداشتن و الامان



پیوست

جزئیات برخی توابع چکیده‌ساز



تابع چکیده‌ساز ساده

□ تابع چکیده‌ساز ساده XOR

■ قطعات داده به عنوان خروجی تابع.

■ اگر داده ورودی m قطعه n بیتی باشد:

$$C_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im} \quad (1 \leq i \leq n)$$

■ احتمال عدم تغییر چکیده در صورت وجود خطای 2^{-n} .

■ در متون عادی، بیت بالای هر بایت معمولاً صفر است (مگر اینکه کاراکتر خاصی باشد که کد اسکی آن بالای ۱۲۸ باشد).

■ در عمل تاثیر این تابع ۱۲۸ بیتی از 2^{128} به 2^{112} کاهش می‌یابد.

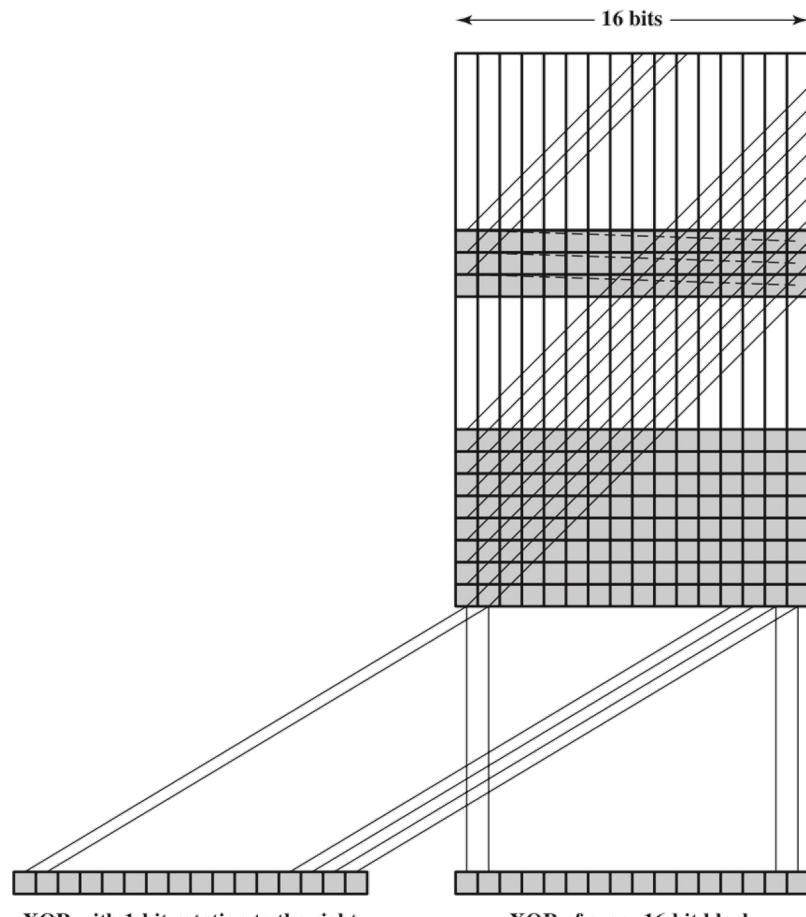


تابع چکیده‌ساز ساده

□ تابع چکیده‌ساز ساده RXOR

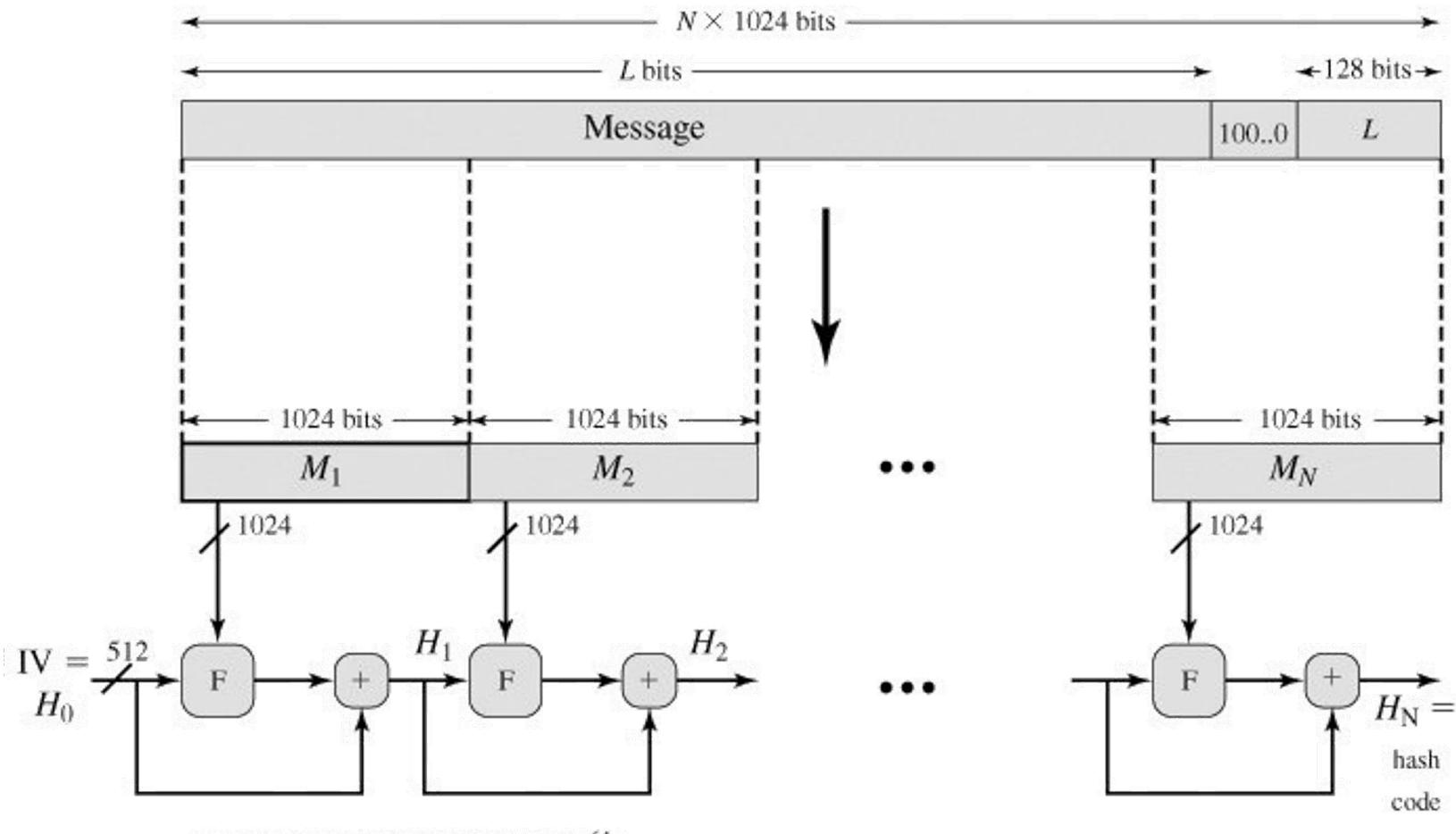
■ در هر مرحله قبل از XOR کردن قطعه جدید با حاصل مراحل قبل، یک شیفت چرخشی تک بیتی به چپ انجام می‌دهد.

■ با توجه به سادگی پیدا کردن تصادم در این تابع، نمی‌توان آن را برای احراز صحت پیام‌هایی که آشکار ارسال می‌شوند (مشابه آنچه که در اسلاید ۳۶ و ۳۷ آمده) استفاده کرد.





الگوریتم SHA-512





مراحل اجرای الگوریتم SHA-512

□ افزودن بیتهای padding

- افزودن ۱۰۰۰...۰ به اندازه‌ای که طول پیام هم‌نهشت با ۸۹۶ شود.

□ افزودن اندازه پیام به انتهای آن

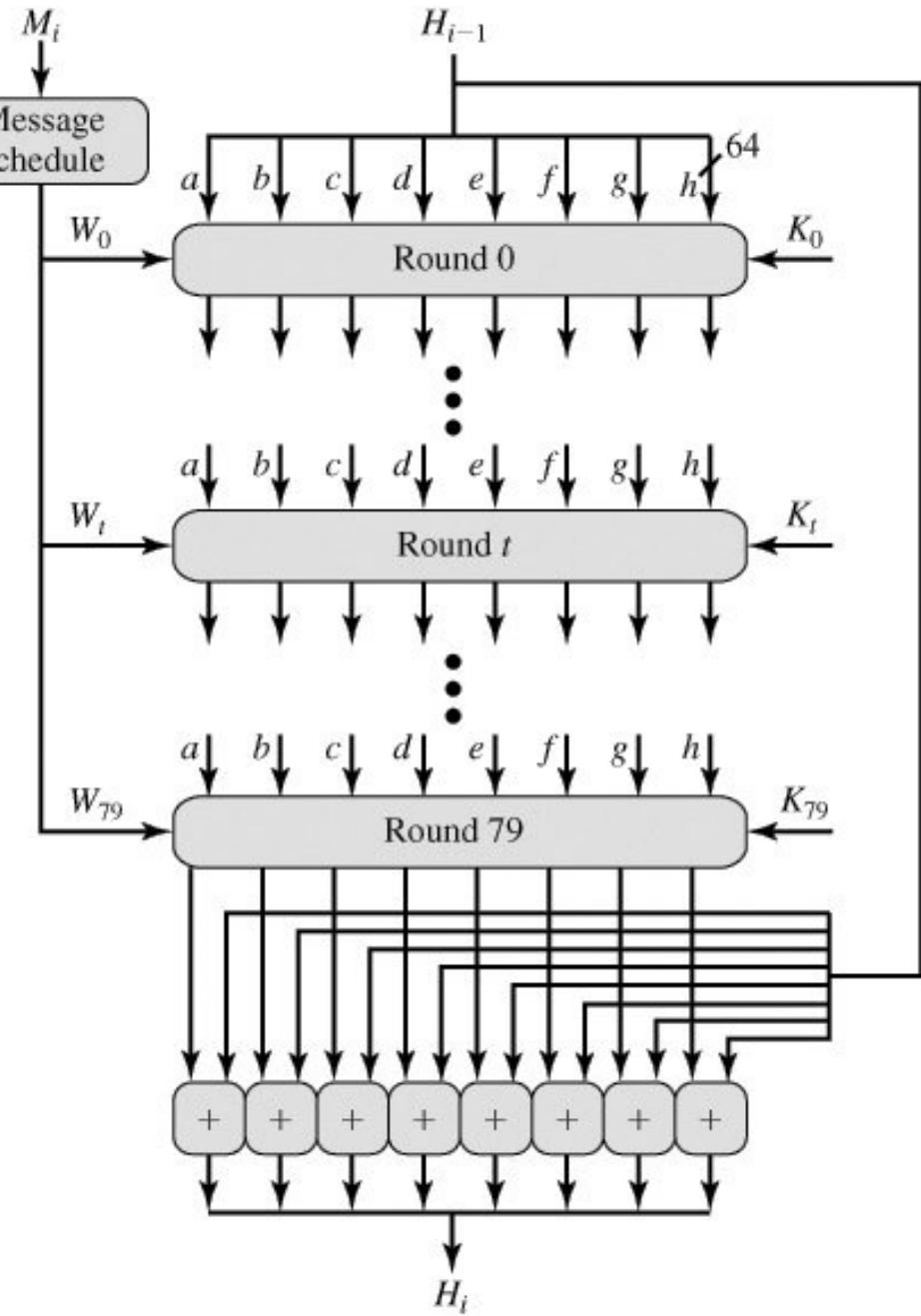
- ثبت طول پیام در ۱۲۸ بیت باقیمانده از قطعه آخر

□ مقداردهی اولیه بافر hash

- مقدار اولیه H_0 در ۸ ثبات ۶۴ بیتی abcdefgh ذخیره می‌شود.

□ پردازش پیام در قطعات ۱۰۲۴ ۱ بیتی (۱۲۸ کلمه‌ای)

- هر قطعه در ۸۰ دور طبق اسلاید بعد پردازش می‌شود.



پردازش یک قطعه در SHA-512

K_i ها ثابت هستند. □

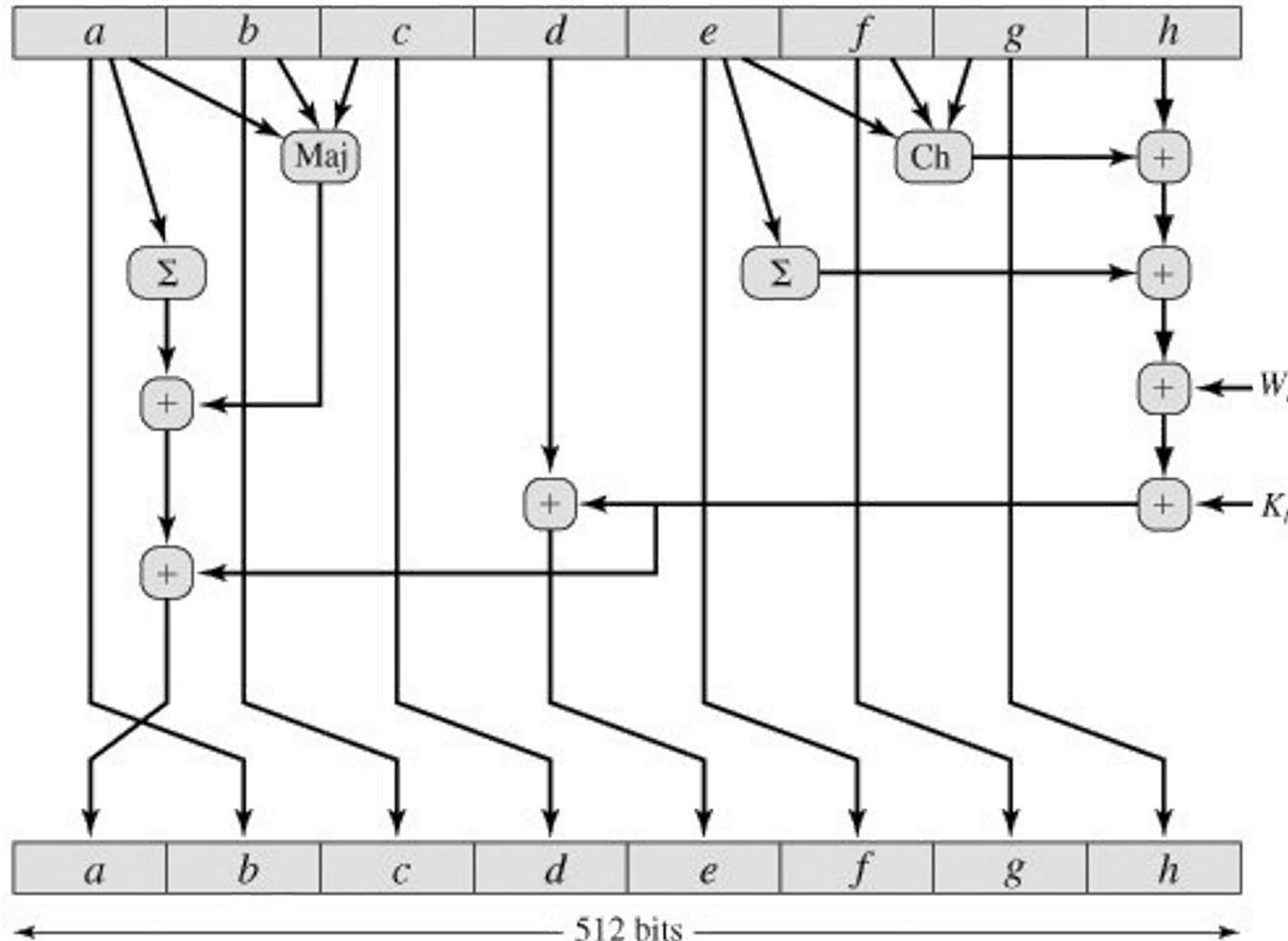
K_i ها شامل ۶۴ بیت اول قسمت □

اعشاری ریشه سوم ۸۰ عدد اول
نخستین هستند. □

W_i های ۶۴ بیتی توسط زمانبند پیام
تولید می شوند. □

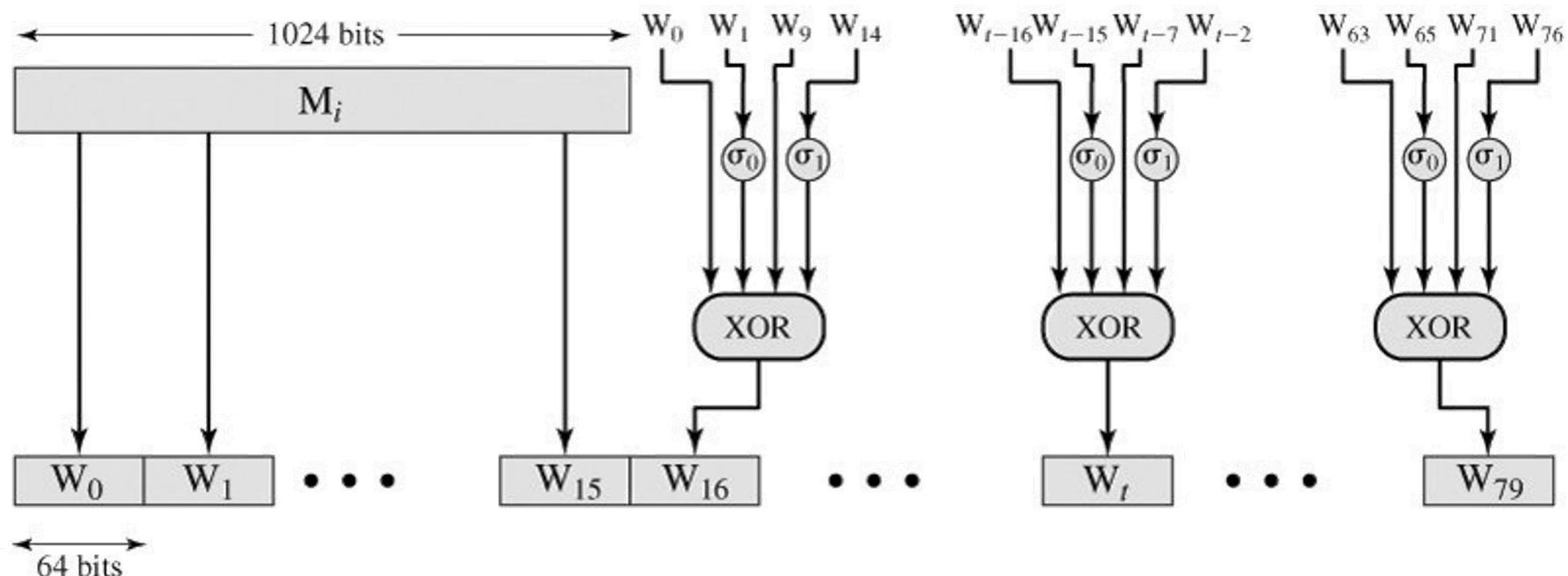


عملیات هر دور در SHA-512





زمانبند پیام در SHA-512





عملکردهای مورد نیاز در SHA-512

$\text{Ch}(e, f, g) = (e \text{ AND } f) \oplus (\text{NOT } e \text{ AND } g)$
the conditional function: If e then f else g

$\text{Maj}(a, b, c) = (a \text{ AND } b) \oplus (a \text{ AND } c) \oplus (b \text{ AND } c)$
the function is true only if the majority (two or three) of the arguments are true

$(\sum_0^{512} a) = \text{ROTR}^{28}(a) \oplus \text{ROTR}^{34}(a) \oplus \text{ROTR}^{39}(a)$

$(\sum_1^{512} e) = \text{ROTR}^{14}(e) \oplus \text{ROTR}^{18}(e) \oplus \text{ROTR}^{41}(e)$

$\text{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

$\sigma_0^{512}(x) = \text{ROTR}^1(x) \oplus \text{ROTR}^8(x) \oplus \text{SHR}^7(x)$

$\sigma_1^{512}(x) = \text{ROTR}^{19}(x) \oplus \text{ROTR}^{61}(x) \oplus \text{SHR}^6(x)$

$\text{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

$\text{SHR}^n(x)$ = left shift of the 64-bit argument x by n bits with padding by zeros on the right

$+$ = addition modulo 2^{64}