یا ذاالامن والامان

کدهای احراز صحت پیام و توابع درهم‌ساز

مبانی امنیت اطلاعات و شبکه‌های کامپیوتری

محمدرضا رازیان*

بهار و تابستان 1395

مرکز تخصصی آپا

دانشگاه سمنان

مرکز تخصصی آپا دانشگاه سمنان

http://cert.semnan.ac.ir

آزمایشگاه امنیت داده و شبکه شریف

http://dnsl.ce.sharif.ir

*Homepage: www.mrazian.com

فهرست مطالب

- مفاهیم اولیه
- رمزگذاری پیام و کدهای تشخیص خطا
- کدهای احراز صحت پیام
- اصول توابع درهم‌ساز
- توابع درهم‌ساز مهم
- HMAC

احراز صحت پیام چیست؟

- اطمینان از:
- صحت محتوای پیام؛ یعنی پیام دریافتی دستکاری نشده است:
  - بدون تغییر
  - بدون درج
  - بدون حذف
- پیام از جانب فرستنده ادعا شده ارسال شده است.
- قابل انکار از سوی منبع (فرستنده) نباشد.

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 سطح کارکرد داریم:
  - سطح اول: استفاده از یک تابع برای تولید عامل احرازکننده
  - سطح دوم: استفاده از یک پروتکل که با استفاده از تابع فوق اصالت پیام را احراز کند.

راهکارهای احراز صحت پیام

- رمزگذاری پیام
  - رمزشده کل پیام به عنوان احراز کننده اصالت پیام
- کد احراز صحت پیام (MAC)
  - تابعی از متن پیام و یک کلید سری (با خروجی با اندازه ثابت) به عنوان احرازکننده پیام
- استفاده از توابع درهم‌ساز برای احراز صحت پیام
  - خروجی حاصل از نگاشت پیام به یک مقدار با طول ثابت (با استفاده از یک تابع درهم ساز) به عنوان احرازکننده پیام

فهرست مطالب

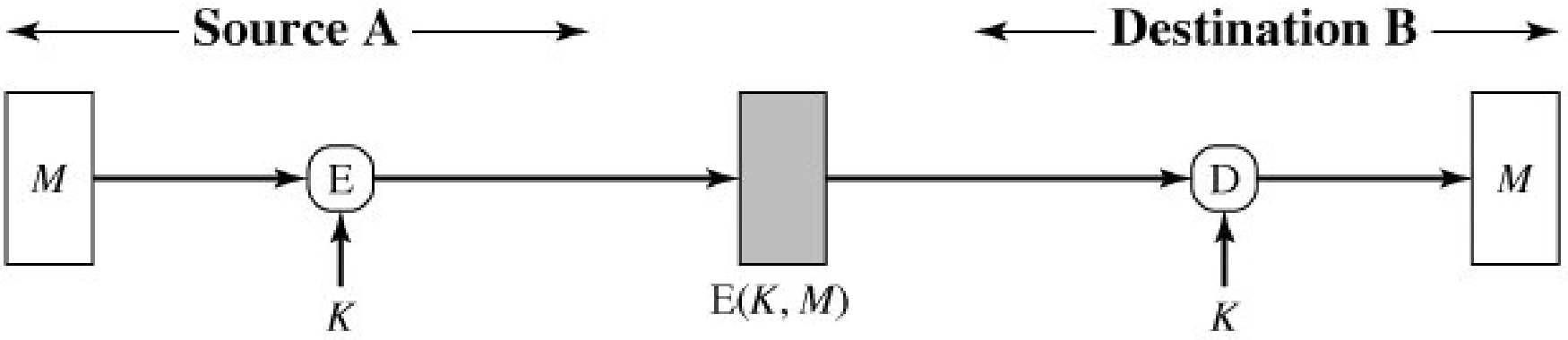
- مفاهیم اولیه
- **رمزگذاری پیام و کدهای تشخیص خطا**
- کدهای احراز صحت پیام
- اصول توابع درهم‌ساز
- توابع درهم‌ساز مهم
- HMAC

رمزگذاری پیام برای احراز صحت پیام

- فرستنده پیام را رمز می‌کند.
- اگر متن رمز شده دستکاری شود با رمزگشایی به متن آشکار نامفهوم (درهم و برهم) می‌رسیم.
- گیرنده، بعد از رمزگشایی چک می‌کند که آیا پیام مفهوم است یا نه؟
- می توان از الگوریتم‌های رمز متقارن و یا نامتقارن برای این منظور استفاده کرد.

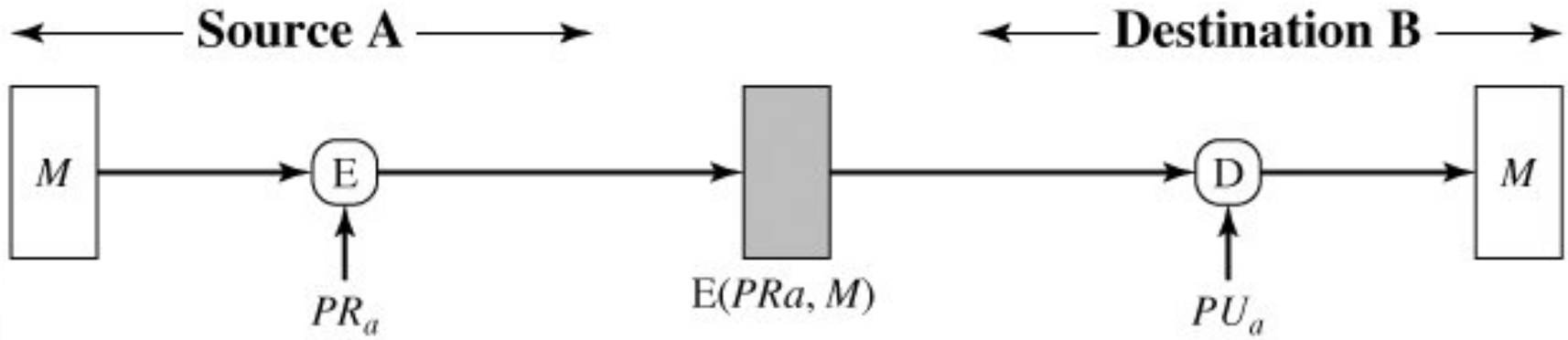
کاربرد رمزگذاری پیام

رمزنگاری متقارن: محرمانگی و احراز صحت



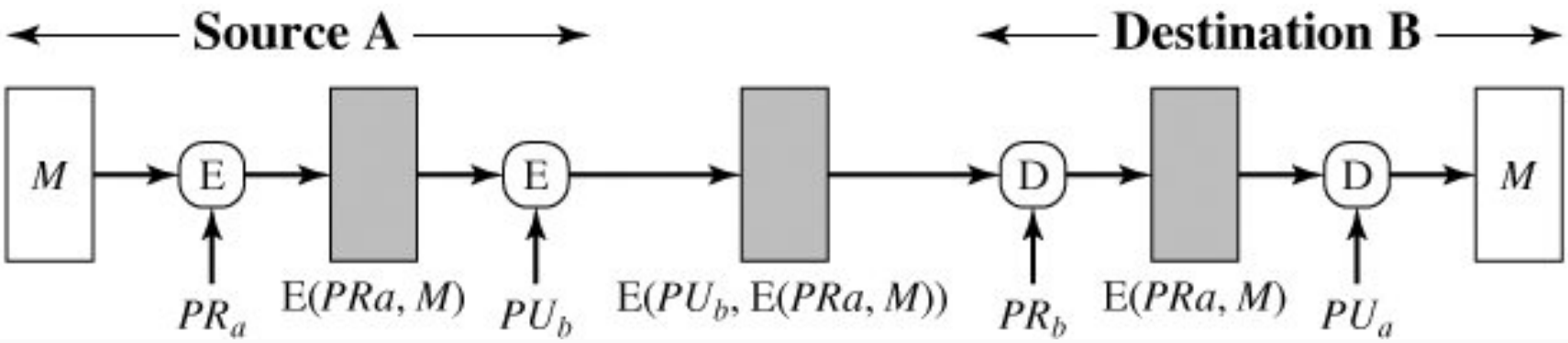
کاربرد رمزگذاری پیام

رمزنگاری کلید عمومی: احراز صحت و امضاء



کاربرد رمزگذاری پیام

رمزنگاری کلید عمومی: محرمانگی، احراز صحت و امضاء



مشکلات رمزنگاری

- بررسی مفهوم بودن محتوا همواره آسان نیست.
  - در حالت کلی با نوعی افزونگی، ساختار درونی مورد انتظار را جستجو می‌کنند.
  - دشواری خودکارسازی فرآیند چک کردن ....
- هنگام ارسال داده
  - اگر داده ها خود تصادفی به نظر برسند، یعنی از ساختار درونی خاصی تبعیت ننمایند، بررسی محتوا تقریباً ناممکن است.
- **راه‌حل اولیه:** استفاده از کدهای تشخیص خطا
  - مثال: یک بیت به انتهای پیام اضافه نماییم، به گونه‌ای که تعداد بیتهای یک، زوج شود.

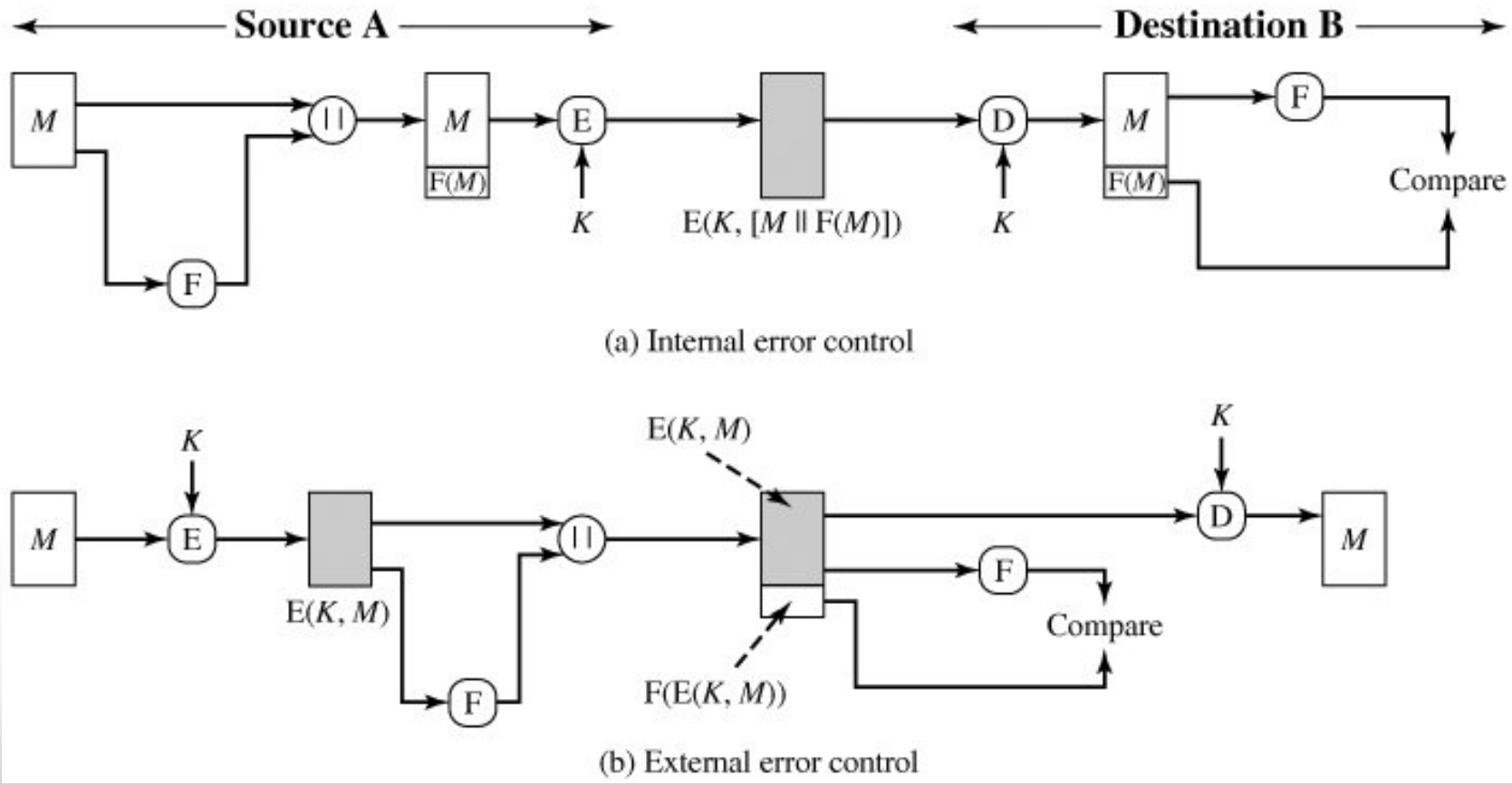
کدهای تشخیص خطا

- تابع F یک کد تشخیص خطا است.
  - اضافه نمودن کدتشخیص خطا (به دنباله بررسی قالب یا FCS- Frame Check Seq نیز معروف است)، توسط تابع F
  - یک مثال از تابع F ، کد CRC است.

- گیرنده، بعد از رمز گشایی چک می‌کند که آیا «کد تشخیص خطای» محاسبه شده توسط F با برچسب پیام مطابقت دارد یا نه؟

انواع کدهای تشخیص خطا

- دو مدل اضافه کردن کد تشخیص خطا (اولی مناسبتر است)



ناامن بودن کدهای تشخیص خطا

- کدهای تشخیص خطا مانند CRC برای تشخیص خطای حاصل از نویز در کاربردهای مخابراتی طراحی شده‌اند.
  - نویز:
    - تغییرات غیرهوشمندانه و غیرعمدی
  - حمله دشمن:
    - تغییرات هوشمندانه و عمدی
- حملات موفقی به الگوریتم‌هایی که از کدهای تشخیص خطا استفاده می‌کردند، صورت پذیرفته است.

نتیجه گیری

- کد تشخیص خطا نمی‌تواند در حالت کلی از دستکاری بسته‌ها جلوگیری کند.
- راه حل: کدهای احراز صحت پیام

فهرست مطالب

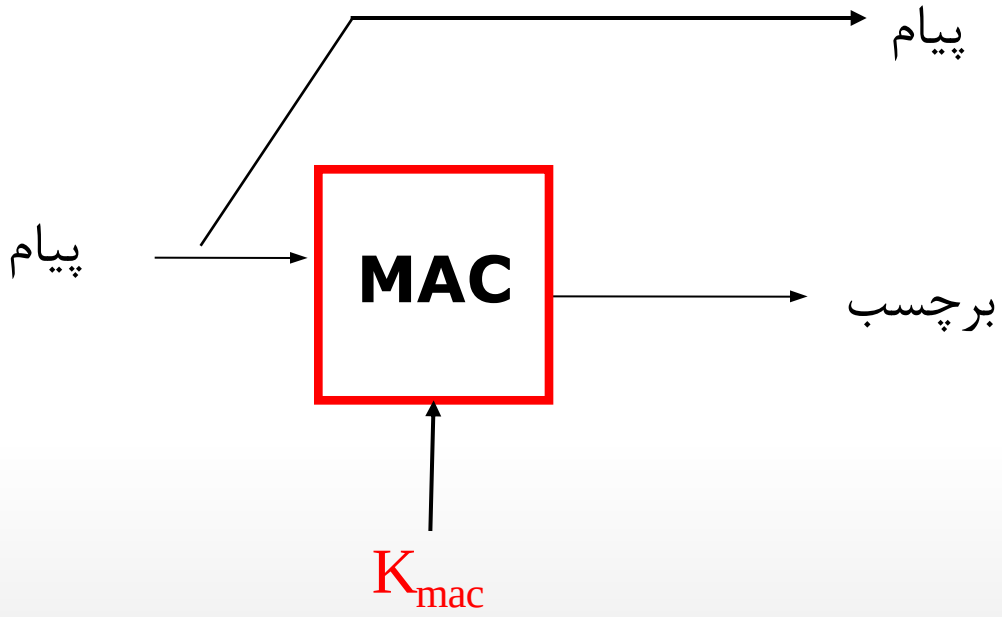
- مفاهیم اولیه
- رمزگذاری پیام و کدهای تشخیص خطا
- **کدهای احراز صحت پیام**
- اصول توابع درهم‌ساز
- توابع درهم‌ساز مهم
- HMAC

کد های احراز صحت پیام

- تولید یک برچسب با طول ثابت:
  - وابسته به پیام
  - لزوماً برگشت پذیر نیست (بر خلاف توابع رمزنگاری)
  - نیازمند اشتراک 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

↓ صحت

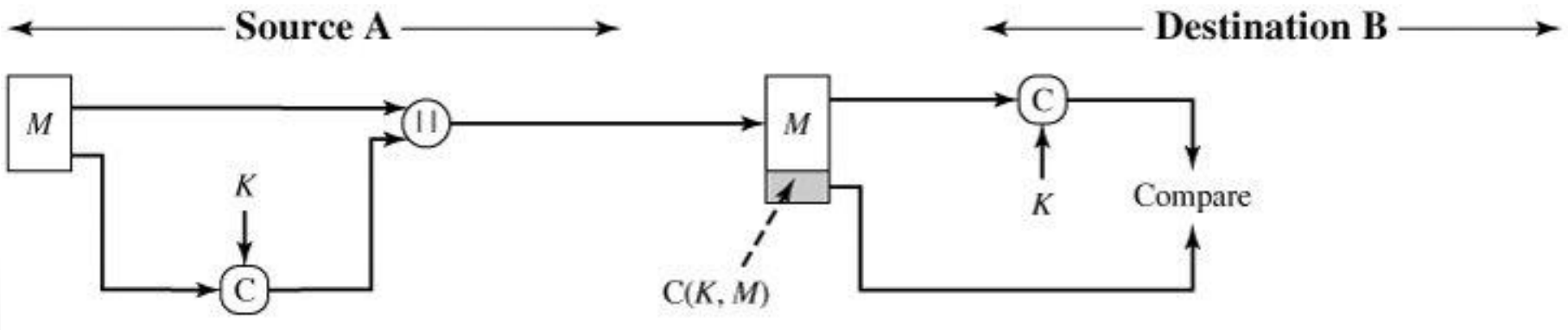


فرق MAC و رمزگذاری

- MAC نیازی ندارد که حتماً برگشت پذیر باشد، در صورتیکه که الگوریتم رمزگذاری باید برگشت‌پذیر باشد.
  - MAC تابع چند به یک است.
  - اندازه 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.

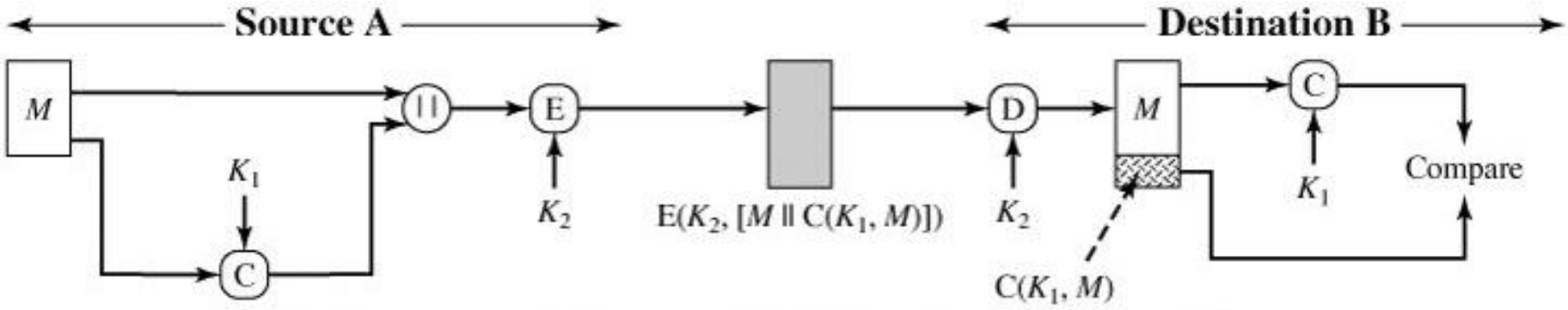
کاربرد کدهای احراز صحت پیام

احراز صحت پیام



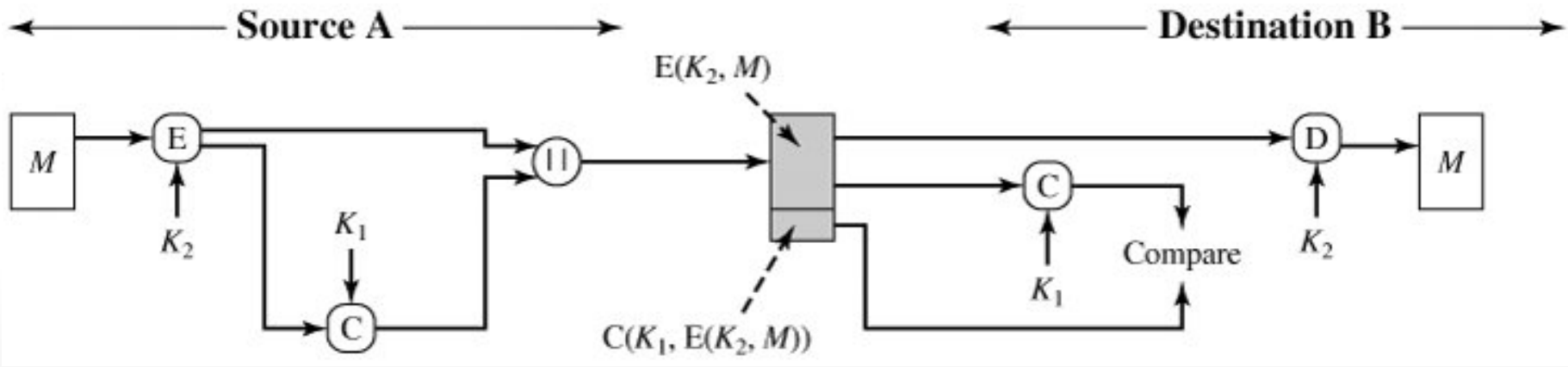
کاربرد کدهای احراز صحت پیام

احراز صحت پیام و محرمانگی؛ احراز صحت پیام آشکار



کاربرد کدهای احراز صحت پیام

احراز صحت پیام و محرمانگی؛ احراز صحت پیام رمز



سوالات متداول در مورد MAC

- چرا از MAC به جای رمزنگاری استفاده می‌کنیم؟
- در بعضی کاربردها نیازی به محرمانگی نداریم...
- در بعضی موقعیت‌ها، قوانین اجازه ارتباط رمزشده را نمی‌دهند...
- اگر از رمزنگاری استفاده نماییم، برای خواندن پیام همیشه به واگشایی رمز نیاز داریم در صورتی که بررسی MAC اختیاری است...
- الگوریتم‌های تولید چکیده پیام عموما از الگوریتم‌های رمزنگاری سریعتر هستند.

سوالات متداول در مورد MAC

- آیا MAC همانند امضا غیر قابل انکار است؟
- خیر
  - امضاء با یک زوج کلید عمومی/خصوصی فراهم می‌شود ولی کلید MAC یک کلید مشترک سری است.
  - بر خلاف امضاء، دو طرف قادر به ایجاد MAC هستند.

امنیت MAC

- حمله آزمون جامع به کلید MAC
  - با داشتن یک متن و MAC آن، به صورت برون (offline) 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 رخ می‌دهد.

امنیت MAC

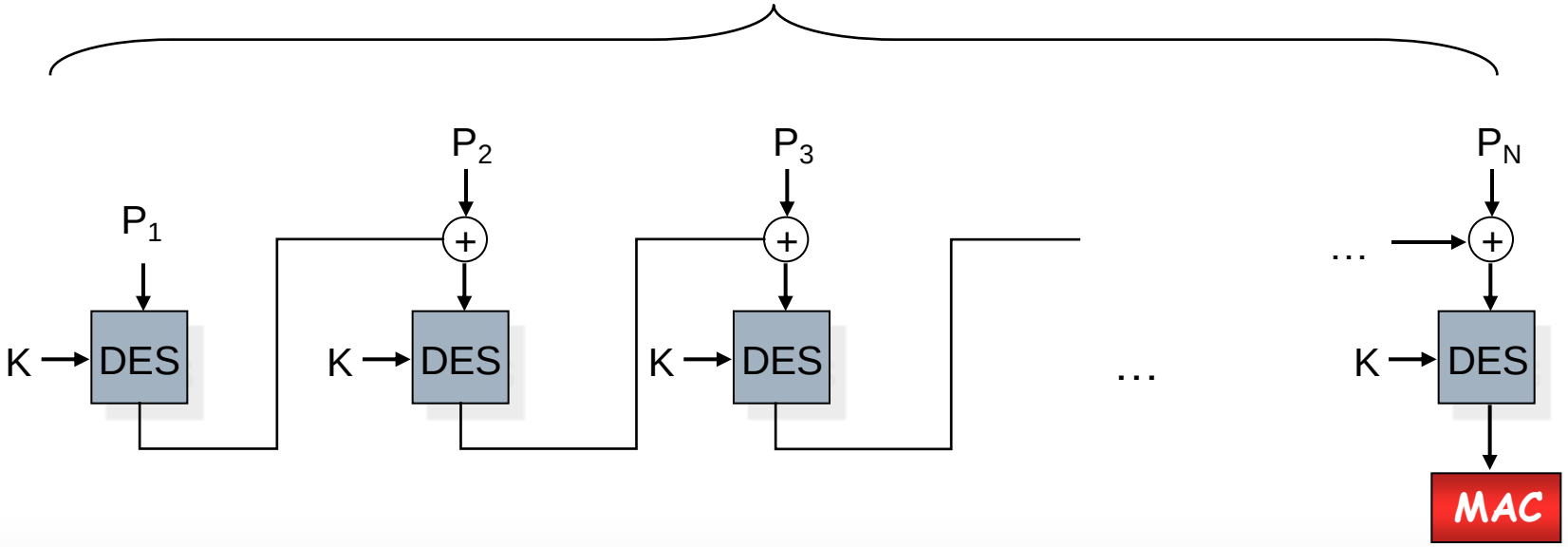
- هزینه لازم برای حمله آزمون جامع به MAC برابر است با
$$\min(2^k, 2^n)$$
- ویژگیهای یک MAC مناسب:
  - با دانستن یک پیام و بر چسب آن، یافتن پیام متفاوتی با برچسب یکسان از لحاظ محاسباتی ناممکن باشد.
  - توزیع خروجی MAC باید یکنواخت باشد تا احتمال اینکه دو پیام تصادفی MAC یکسان داشته باشند، کمینه شود.
- نکته: طول برچسب MAC همانند طول کلید در امنیت MAC تاثیر دارد.

کداحراز صحت پیام DAA

- DAA (Data Authentication Algorithm)
- استاندارد NIST و ANSI X9.17
- بر اساس رمز قطعه‌ای DES و مد کاری CBC
- همانند رمز نگاری CBC، پیام را پردازش کرده و تنها آخرین قطعه را به عنوان برچسب استفاده می‌کنیم.

DAA

متن آشکار (تقسیم شده به قطعات)



فهرست مطالب

- مفاهیم اولیه
- رمزگذاری پیام و کدهای تشخیص خطا
- کدهای احراز صحت پیام
- **اصول توابع درهم‌ساز**
- توابع درهم‌ساز مهم
- HMAC

توابع درهم‌ساز

- 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 درهم‌ساز–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 درهم‌ساز

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

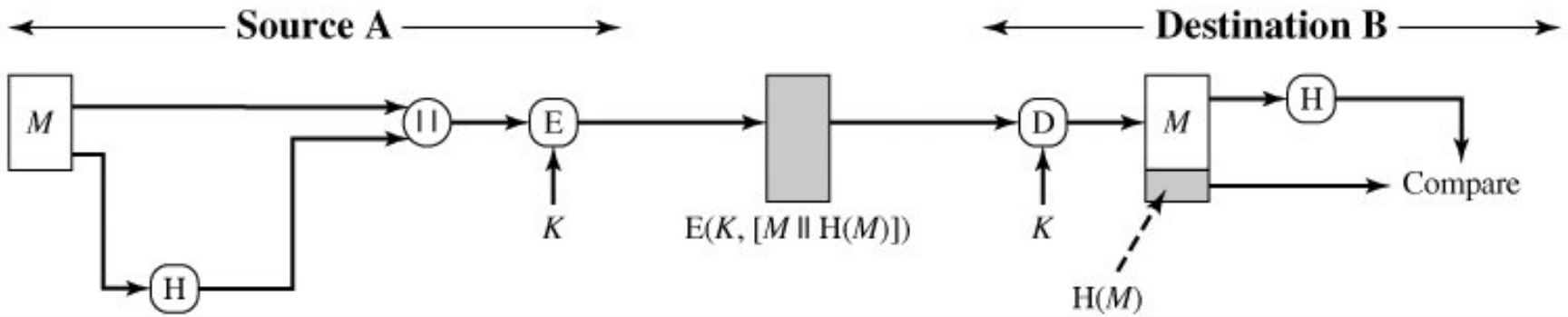
امنیت توابع درهم‌ساز

- توابع درهم ساز باید یک طرفه باشند.
  - پیچیدگی جستجوی کامل (آزمون جامع) 2^n است، که n طول خروجی تابع است.
- مقاومت در برابر تصادم (ضعیف)
  - پیچیدگی جستجوی کامل (آزمون جامع) 2^n است.
- مقاومت در برابر تصادم (قوی)
  - پیچیدگی جستجوی کامل (آزمون جامع) $2^{n/2}$ 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.

با کمک حمله روز تولد

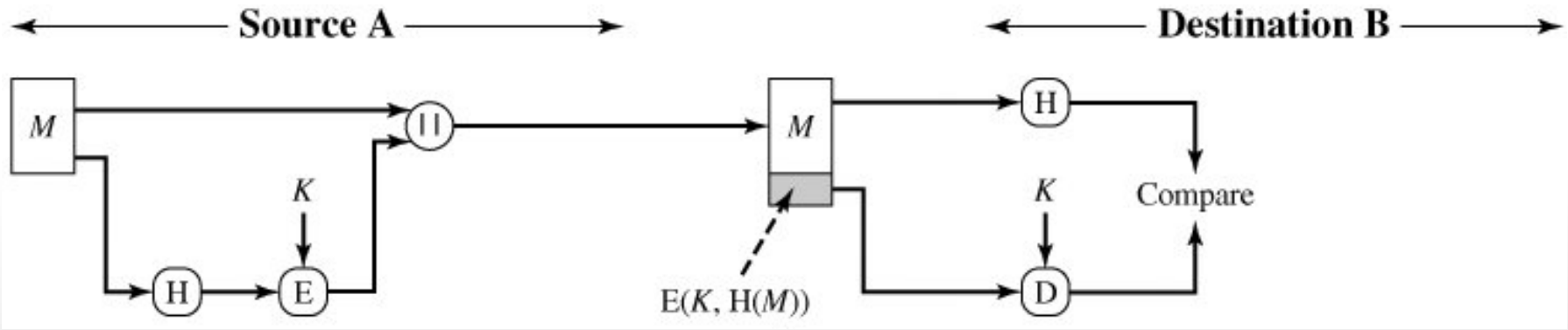
کاربرد توابع درهم‌ساز

احراز صحت پیام و محرمانگی در ترکیب با رمز متقارن



کاربرد توابع درهم‌ساز

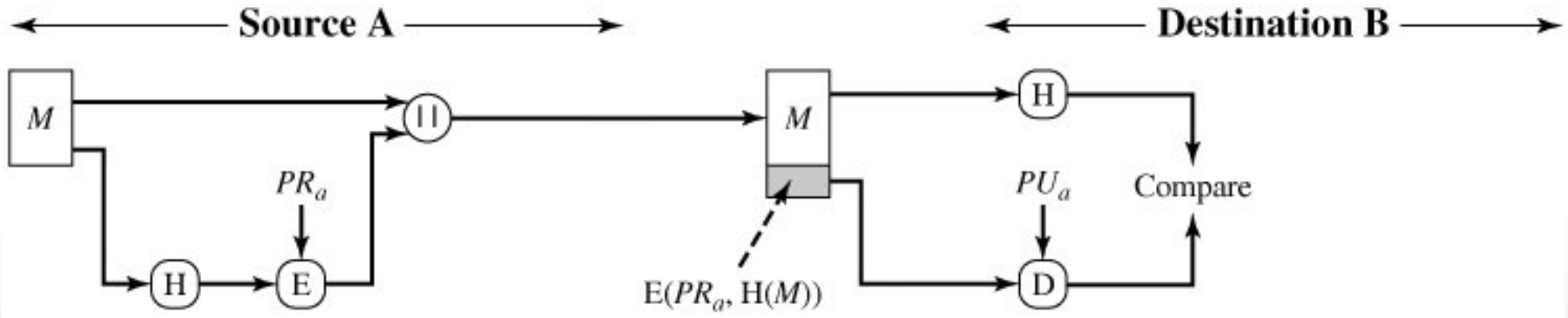
- **احراز صحت پیام در ترکیب با رمز متقارن**
  - صرفاً رمزگذاری چکیده پیام
  - این ترکیب در واقع یک کد احراز پیام را می‌سازد.



کاربرد توابع درهم‌ساز

احراز صحت پیام در ترکیب با رمز کلید عمومی

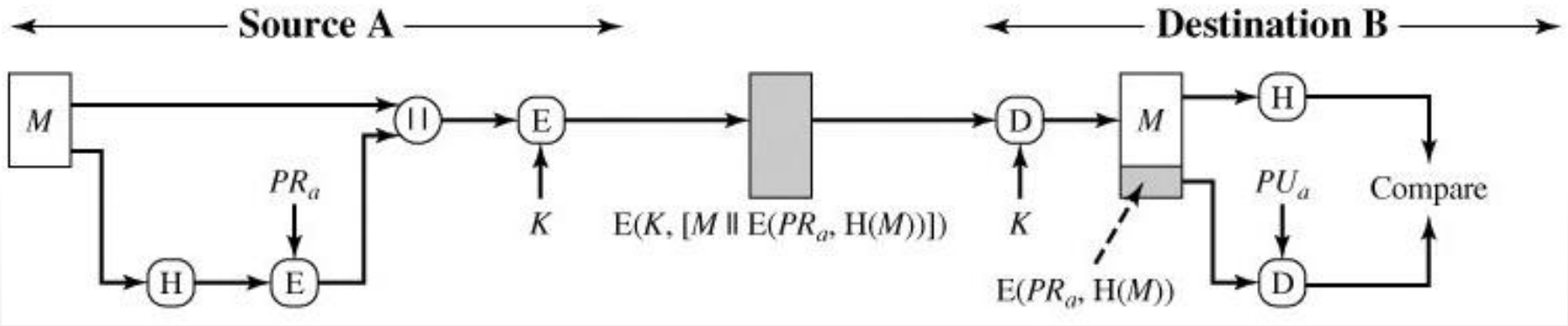
- صرفاً رمزگذاری چکیده پیام
- این ترکیب در واقع یک امضای رقمی را می‌سازد.



کاربرد توابع درهم‌ساز

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

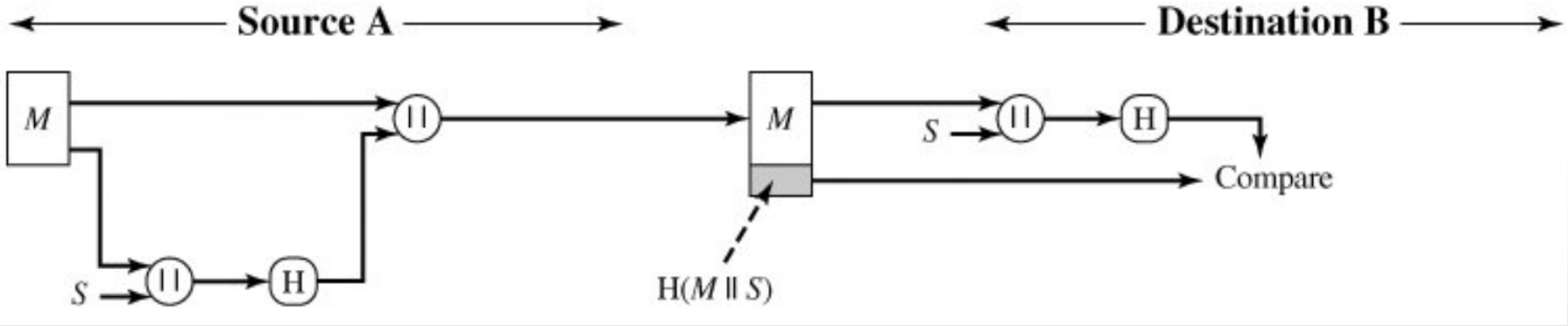
- امضای رقمی با رمز نامتقارن برای حفظ صحت
- رمز متقارن برای حفظ محرمانگی



کاربرد توابع درهم‌ساز

احراز صحت پیام بدون رمزگذاری

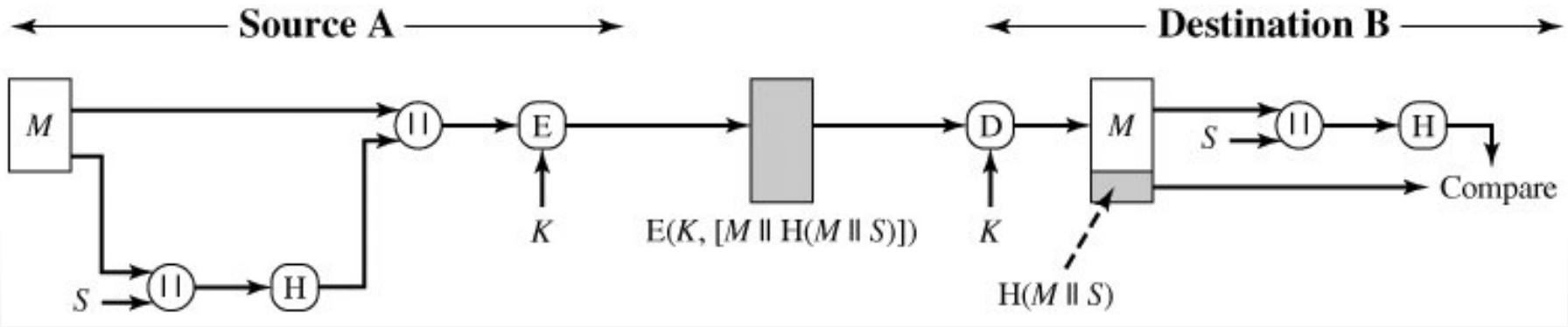
- طرفین راز S را مخفیانه به اشتراک می‌گذارند.
- بدون استفاده از رمزگذاری
- کاربرد عملی زیاد



کاربرد توابع درهم‌ساز

احراز صحت پیام بدون رمزگذاری و محرمانگی با رمز متقارن

- رمزگذاری صرفاً برای محرمانگی



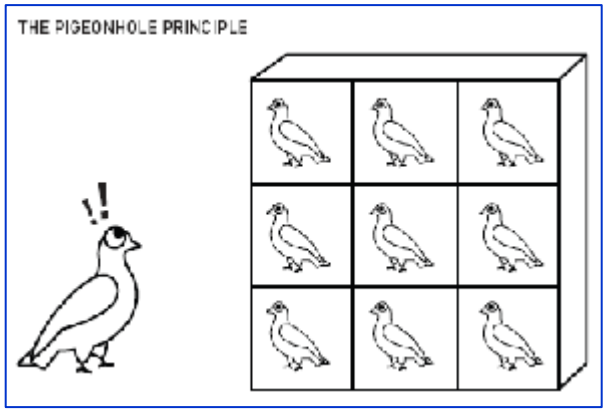
مقایسه رمزنگاری و توابع درهم‌ساز

- رمزهای قطعه‌ای:
  - از لحاظ اجرایی، پیاده‌سازی نرم‌افزاری رمزهای قطعه‌ای کندتر از توابع درهم ساز
  - دارای هزینه سخت‌افزاری بیشتر
  - کارایی کمتر برای بلاک‌های داده‌ای حجیم
  - دارای محدودیت‌های صادراتی (Export Control)

پارادوکس روز تولد

- در میان ۲۳ نفر، احتمال یافتن دو نفر که در یک روز از سال متولد شده اند بیش از ۵۰٪ است.

- این پارادوکس به دو شکل عمومی توسعه یافت.
  - اندازه حداقل یک مجموعه برای یافتن یک زوج تکراری در آن با احتمال بیش از ۰/۵
  - اندازه حداقل دو مجموعه برای یافتن یک تصادم بین اعضای آنها با احتمال بیش از ۰/۵



$$P(n) = \frac{۳۶۵}{۳۶۵} \times \frac{۳۶۵ - ۱}{۳۶۵} \times \frac{۳۶۵ - ۲}{۳۶۵} \times \ldots \times \frac{۳۶۵ - n + ۱}{۳۶۵}$$

$$Q(۱) = ۰/۰۰۰۰$$

$$Q(۲) \approx ۰/۰۰۲۷$$

$$Q(۳) \approx ۰/۰۰۸۲$$

$$\vdots$$

$$Q(۶) \approx ۰/۰۴۰۵$$

$$\vdots$$

$$Q(۱۰) \approx ۰/۱۱۶۵$$

$$\vdots$$

$$Q(۱۵) \approx ۰/۲۵۲۹$$

$$\vdots$$

$$Q(۲۲) \approx ۰/۴۷۵۷$$

$$\mathbf{Q(۲۳) \approx ۰/۵۰۷۳}$$

$$\vdots$$

$$Q(۳۵) \approx ۰/۸۱۴۴$$

$$Q(۵۰) \approx ۰/۹۷۰۴$$

پارادوکس روز تولد

- **مبنای 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 ۰/۵ است.

حمله روز تولد

- ممکن است تصور کنید یک MAC یا 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 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 زوج را امضا نماید، و سپس پیام دلخواه دشمن را جایگزین می‌کنیم.

{ This letter is / I am writing } to introduce { you to / to you } { Mr. / -- } Alfred { P. / -- }

Barton, the { new / newly appointed } { chief / senior } jewellery buyer for { our / the }

Northern { European / Europe } { area / division }. He { will take / has taken } over { the / -- }

responsibility for { all / the whole of } our interests in { watches and jewellery / jewellery and watches }

in the { area / region }. Please { afford / give } him { every / all the } help he { may need / needs }

to { seek out / find } the most { modern / up to date } lines for the { top / high } end of the

market. He is { empowered / authorized } to receive on our behalf { samples / specimens } of the

{ latest / newest } { watch and jewellery / jewellery and watch } products, { up / subject } to a { limit / maximum }

of ten thousand dollars. He will { carry / hold } a signed copy of this { letter / document }

as proof of identity. An order with his signature, which is { appended / attached }

{ authorizes / allows } you to charge the cost to this company at the { above / head office }

address. We { fully / -- } expect that our { level / volume } of orders will increase in

the { following / next } year and { trust / hope } that the new appointment will { be / prove }

{ advantageous / an advantage } to both our companies.

Figure 11.6 A Letter in 2^{37} Variation [DAVI89]

مثالی از حمله (نمونه 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 دشمن)

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.

توابع درهم‌ساز ساده

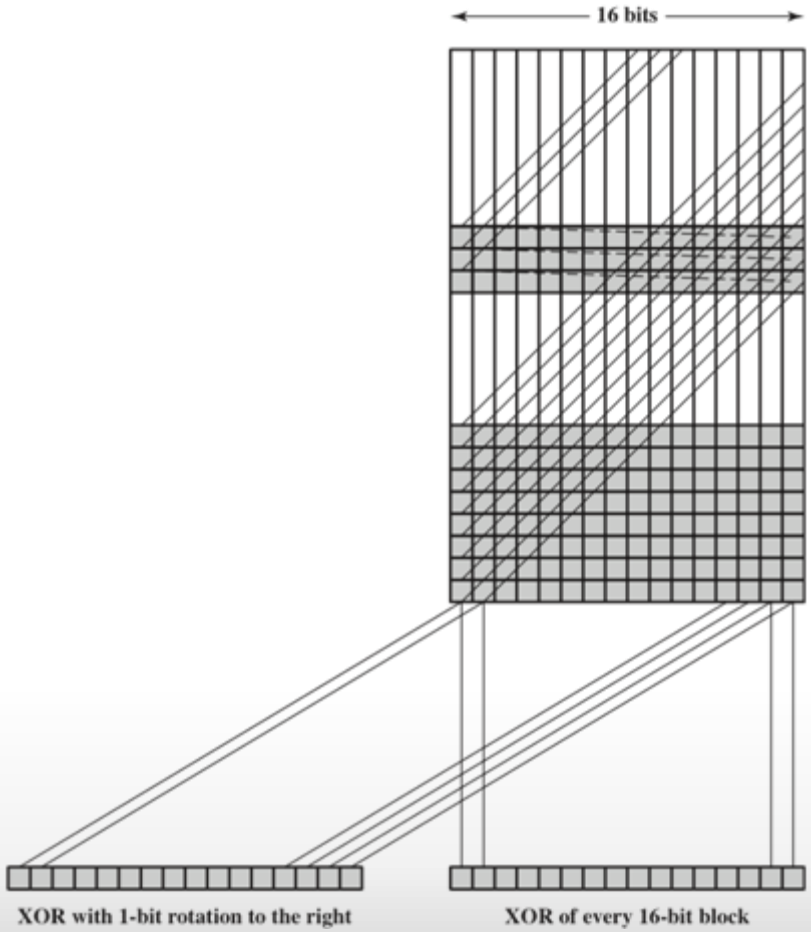
- **تابع درهم‌ساز ساده XOR**
- XOR قطعات داده به عنوان خروجی تابع.
- اگر داده ورودی m قطعه n بیتی باشد:

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

- در متون عادی، بیت بالای هر بایت معمولا صفر است (مگر اینکه کاراکتر خاصی باشد که کد اسکی آن بالای ۱۲۸ باشد).
- درعمل تاثیر این تابع ۱۲۸ بیتی از 2^{128} به 2^{112} کاهش می‌یابد و با حمله روز تولد به 2^{56} کاهش می یابد.

توابع درهم‌ساز ساده

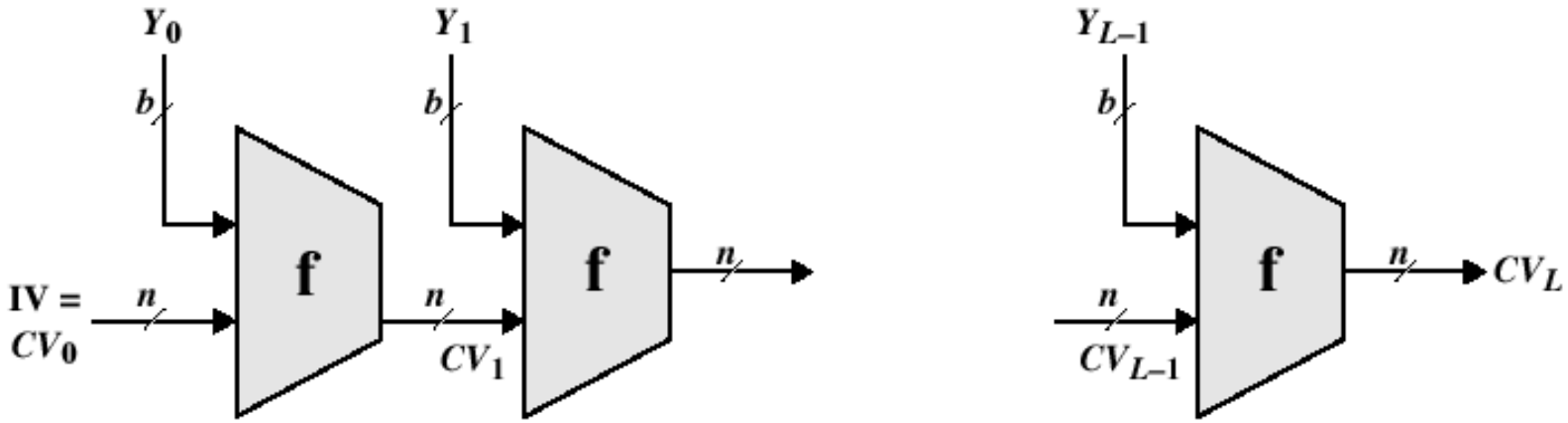
- **تابع درهم‌ساز ساده RXOR**
  - در هر مرحله قبل از XOR کردن قطعه جدید با حاصل مراحل قبل، یک شیفت چرخشی تک بیتی به چپ انجام می‌دهد.
  - با توجه به سادگی پیدا کردن تصادم در این تابع، نمی‌توان آن را برای احراز صحت پیام‌هایی که آشکار ارسال می‌شوند (مشابه آنچه که در اسلاید ۳۶ و ۳۷ آمده) استفاده کرد.



ساختار درونی توابع درهم‌ساز

- اِعمال مکرر یک تابع فشرده‌ساز به یک رشته با طول ثابت
- اگر تابع فشرده‌ساز مقاوم در برابر تصادم باشد، تابع درهم‌ساز نیز همین‌گونه خواهد بود.
- توابع معروفی مانند
  - MD5: Message Digest 5
  - SHA-1: Secure Hash Algorithm -1
- از همین ایده استفاده می‌کنند.

ساختار درونی توابع درهم ساز



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

فهرست مطالب

- مفاهیم اولیه
- رمزگذاری پیام و کدهای تشخیص خطا
- کدهای احراز صحت پیام
- اصول توابع درهم‌ساز
- **توابع درهم‌ساز مهم**
- HMAC

توابع درهم ساز مهم: 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 (قوی) تحت حمله آزمون جامع: 2^{64}
- امروزه امن محسوب نمی‌شود.

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

توابع درهم ساز مهم: SHA-1

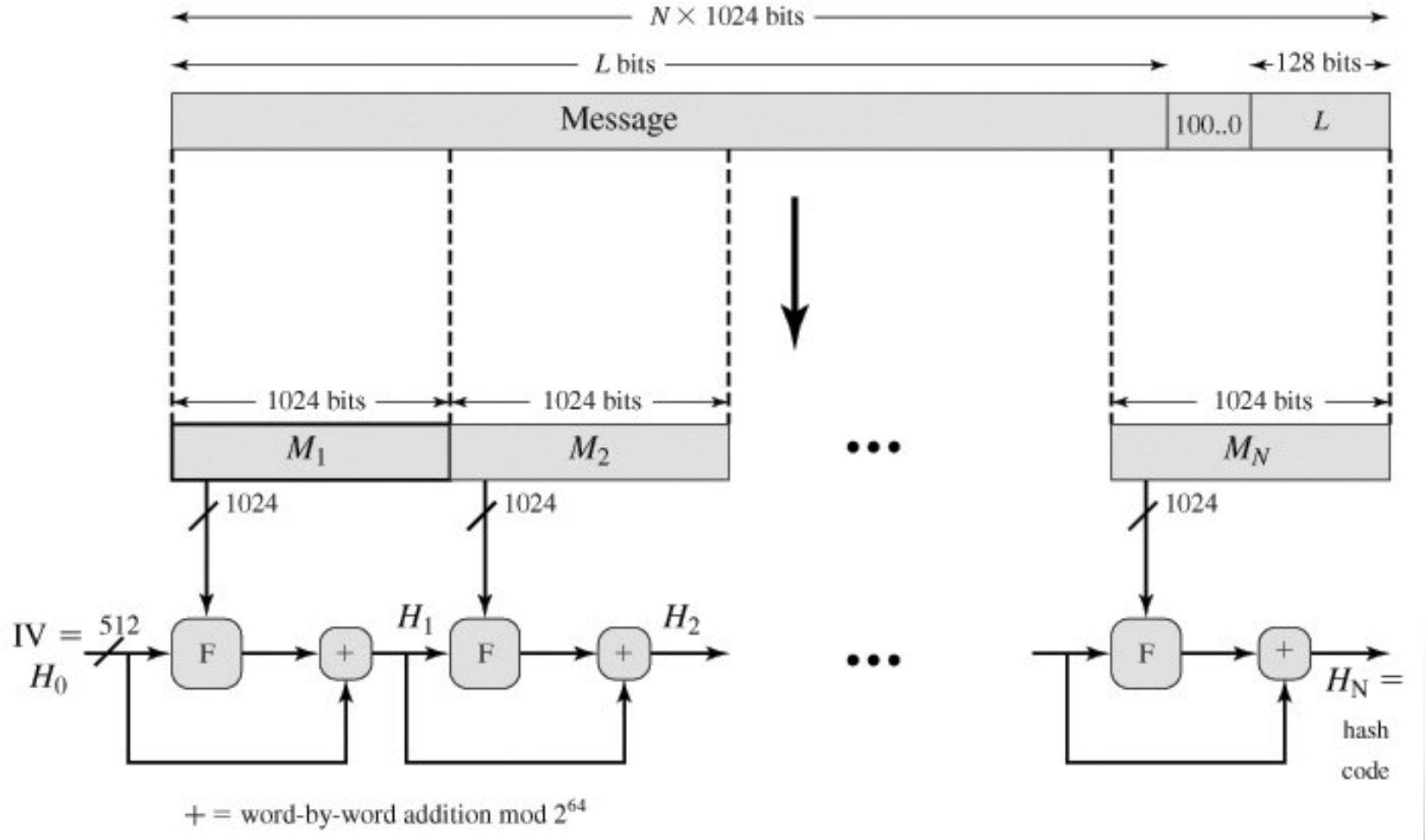
- SHA-1: Secure Hash Algorithm – 1
  - استاندارد NIST، ۱۹۹۵
  - طول ورودی کوچکتر از $۲^{۶۴}$ بیت
  - طول خروجی ۱۶۰ بیت
  - استفاده شده در استاندارد امضای دیجیتال DSS
- امنیت:
  - مقاومت در برابر تصادم (قوی) تحت حمله روز تولد: $۲^{۸۰}$
  - در سال ۲۰۰۵ توانستند با $۲^{۶۹}$ عمل یک تصادم در آن بیابند.
  - در آمریکا از سال ۲۰۱۰ باید با گونه‌های امن‌تر آن یعنی خانواده SHA-2 جایگزین شده باشد.

توابع درهم ساز مهم: SHA-2

- نسخه‌های زیر نیز علاوه بر SHA-1 استاندارد شده اند:
  - SHA-256، SHA-384 و SHA-512
  - معروف به خانواده SHA-2 هستند.
  - از لحاظ ساختار و جزئیات مشابه SHA-1 هستند.

Algorithm	Digest size	Block size	Message size	Security
SHA-1	160	512	$< 2^{64}$	80 bits
SHA-256	256	512	$< 2^{64}$	128 bits
SHA-384	384	1024	$< 2^{128}$	192 bits
SHA-512	512	1024	$< 2^{128}$	256 bits

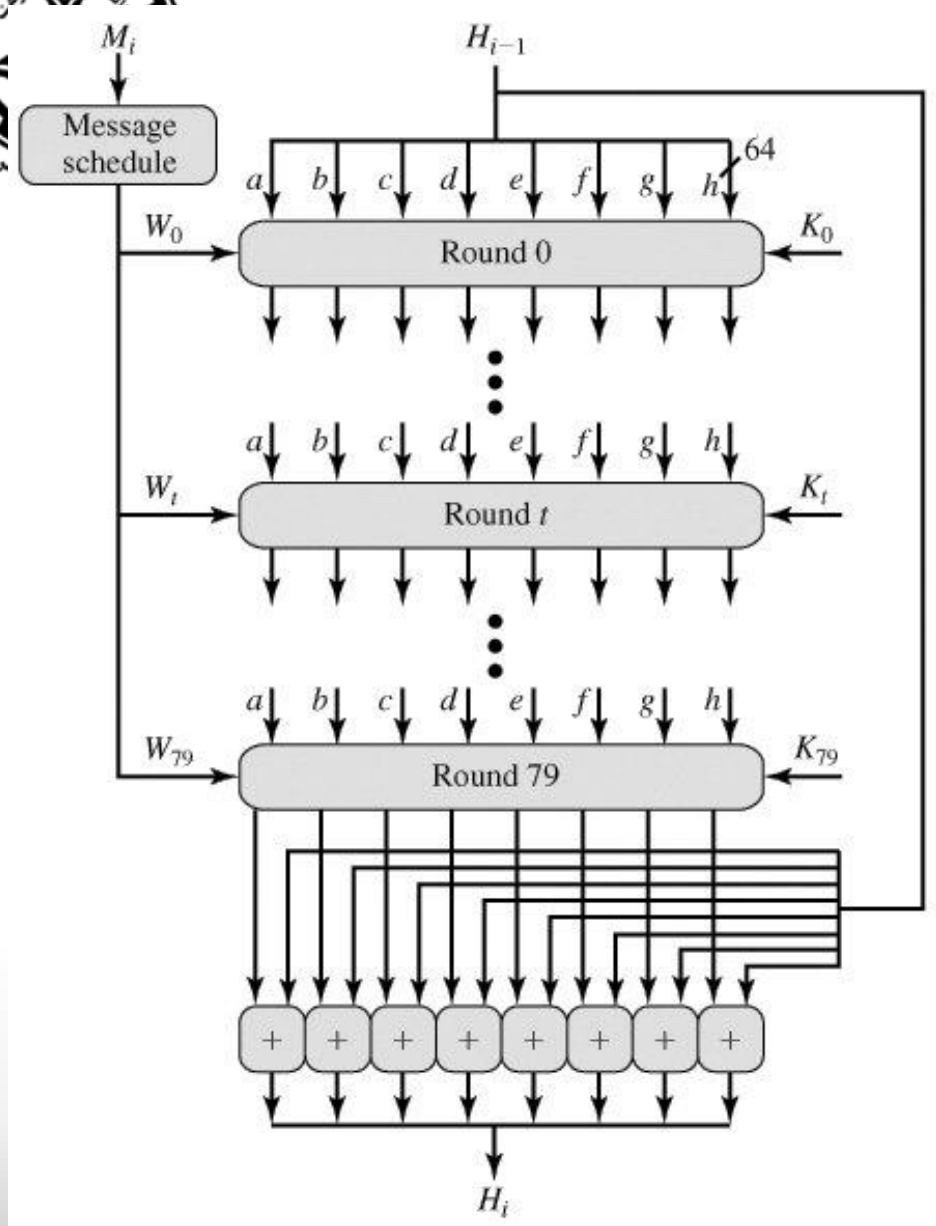
الگوریتم SHA-512



مراحل SHA-512

- افزودن بیتهای padding
  - افزودن 1000...0 به اندازه ای 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 بیتی (۱۲۸ کلمه‌ای)
  - هر قطعه در ۸۰ دور طبق اسلاید بعد پردازش می شود.

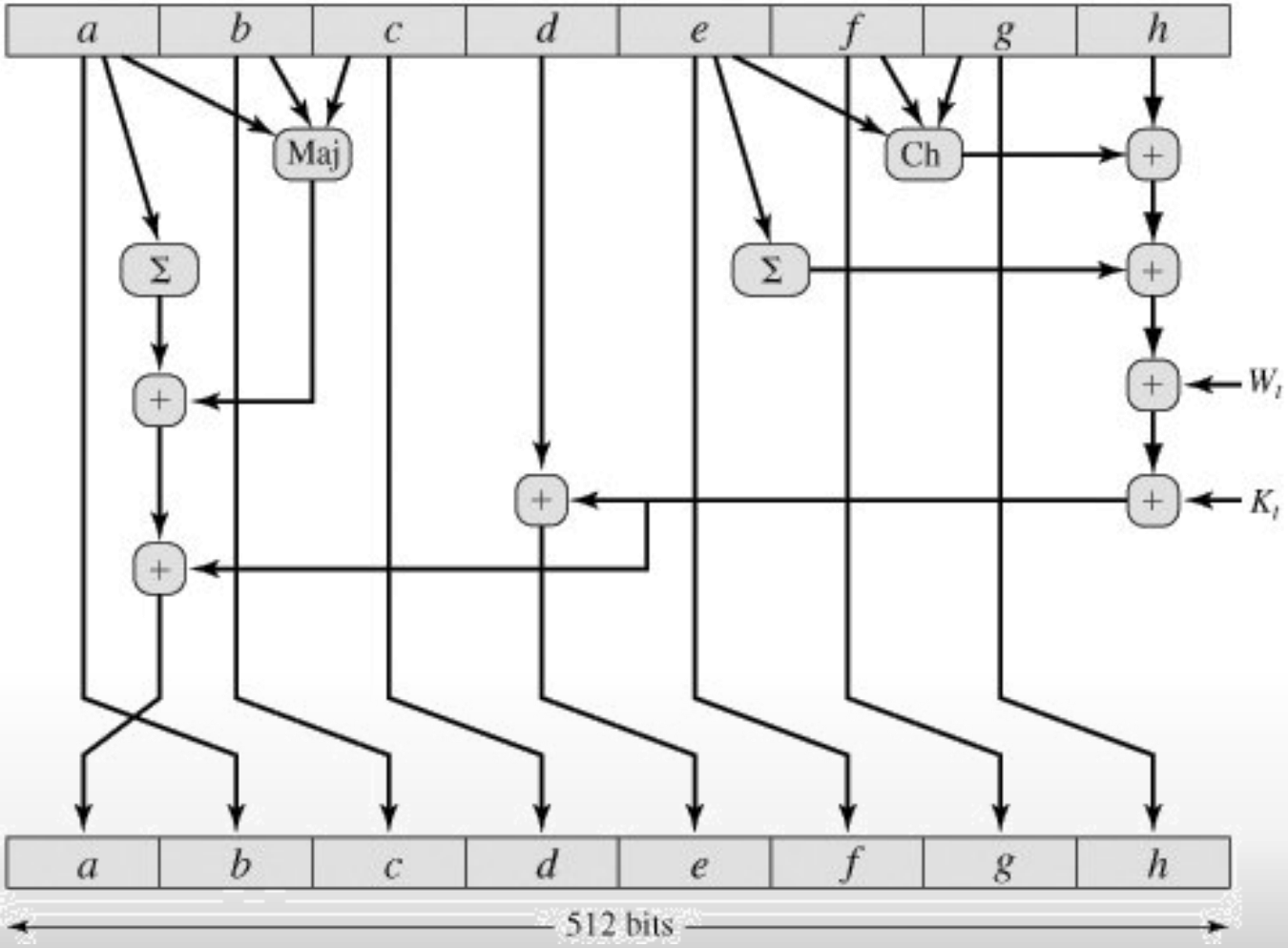
۶۴ بیت اول قسمت اعشاری جذر ۸ عدد اول نخستین



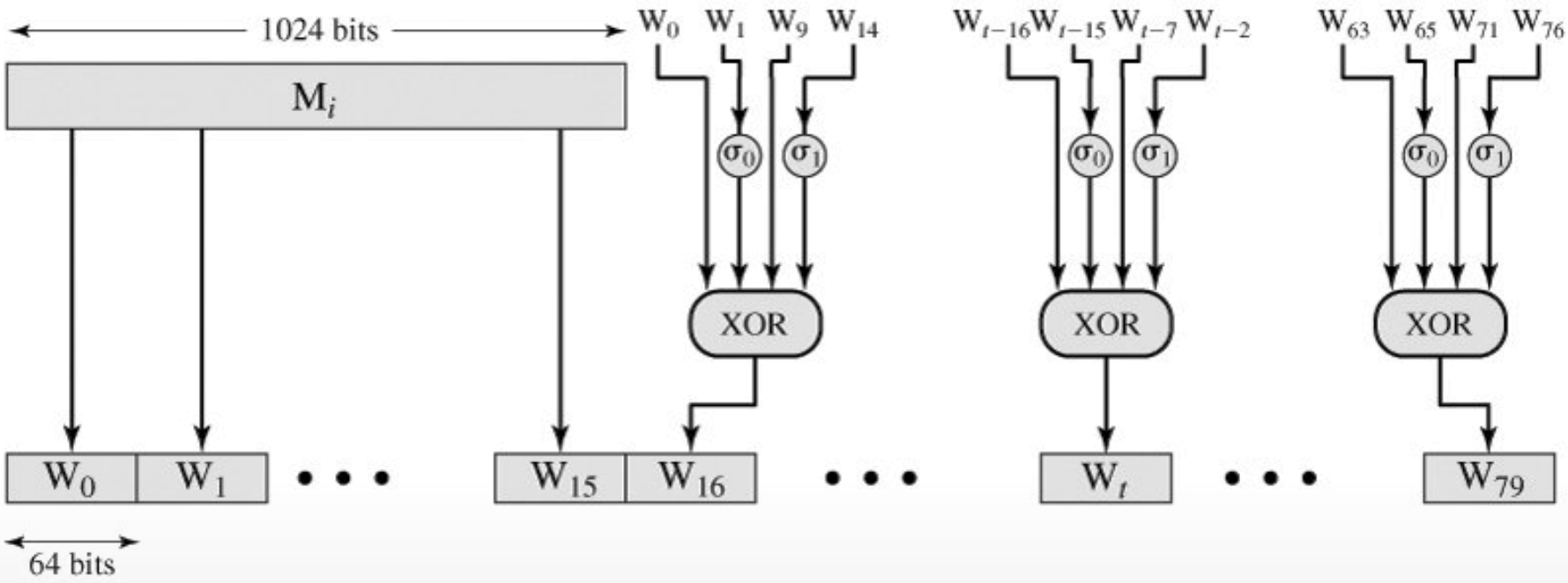
پردازش یک قطعه در SHA-512

- K_iها ثابت هستند.
- K_iها شامل ۶۴ بیت اول قسمت اعشاری ریشه سوم ۸۰ عدد اول نخستین هستند.
- W_iهای ۶۴ بیتی توسط زمانبند پیام تولید می شوند.

عملیات هر دور در SHA-512



زمانبند پیام در SHA-512



عملگرهای مورد نیاز در SHA-512

$$\begin{aligned}
\mathrm{Ch}(e, f, g) &= (e \text{ AND } f) \oplus (\text{NOT } e \text{ AND } g) \\
&\quad \textit{the conditional function: If } e \textit{ then } f \textit{ else } g \\
\mathrm{Maj}(a, b, c) &= (a \text{ AND } b) \oplus (a \text{ AND } c) \oplus (b \text{ AND } c) \\
&\quad \textit{the function is true only of the majority (two or three) of the arguments are true} \\
\left(\sum_0^{512} a\right) &= \mathrm{ROTR}^{28}(a) \oplus \mathrm{ROTR}^{34}(a) \oplus \mathrm{ROTR}^{39}(a) \\
\left(\sum_1^{512} e\right) &= \mathrm{ROTR}^{14}(e) \oplus \mathrm{ROTR}^{18}(e) \oplus \mathrm{ROTR}^{41}(e) \\
\mathrm{ROTR}^n(x) &= \text{circular right shift (rotation) of the 64-bit argument } x \text{ by } n \text{ bits}
\end{aligned}$$

$$\begin{aligned}
\sigma_0^{512}(x) &= \mathrm{ROTR}^1(x) \oplus \mathrm{ROTR}^8(x) \oplus \mathrm{SHR}^7(x) \\
\sigma_1^{512}(x) &= \mathrm{ROTR}^{19}(x) \oplus \mathrm{ROTR}^{61}(x) \oplus \mathrm{SHR}^6(x) \\
\mathrm{ROTR}^n(x) &= \text{circular right shift (rotation) of the 64-bit argument } x \text{ by } n \text{ bits} \\
\mathrm{SHR}^n(x) &= \text{left shift of the 64-bit argument } x \text{ by } n \text{ bits with padding by zeros on the right} \\
+ &= \text{addition modulo } 2^{64}
\end{aligned}$$

تابع درهم‌ساز SHA-3

- به دلایل زیر NIST در سال ۲۰۰۷ مسابقه‌ای را برای انتخاب تابع درهم‌ساز جدید و معرفی آن به عنوان تابع استاندارد آغاز کرد.
  - توابع SHA-1 و SHA-2 از ساختاری مشابه ساختار MD5 و SHA-0 (که شکسته شده‌اند) استفاده می‌نمایند.
  - به صورت نظری حملاتی به SHA-1 و SHA-2 وارد است (هر چند که این حملات در عمل قابلیت اجرا ندارند).
- در سال ۲۰۱۲ تابع درهم‌ساز Keccak به عنوان برنده و تابع SHA-3 تعیین گردید.
- در حال حاضر پیش‌نویس استاندارد SHA-3 منتشر شده است و ویرایش نسخه نهایی آن هنوز به پایان نرسیده است.

فهرست مطالب

- مفاهیم اولیه
- رمزگذاری پیام و کدهای تشخیص خطا
- کدهای احراز صحت پیام
- اصول توابع درهم‌ساز
- توابع درهم‌ساز مهم
- **HMAC**

کد احراز اصالت HMAC

- HMAC یک الگوریتم احراز صحت پیام است.
- HMAC اساساً روشی برای ترکیب کردن کلید مخفی با الگوریتم‌های درهم‌ساز فعلی است.
- برای تولید چکیده پیام، از توابع درهم‌ساز استفاده شده است.
  - در مقابل استفاده از رمزهای قطعه‌ای
  - بدلیل مزایای عملی توابع درهم ساز

کد احراز اصالت HMAC

- HMAC جزو ملزومات پیاده‌سازی IPSec است.
- HMAC به طور گسترده استفاده می‌شود (مثلاً SSL).

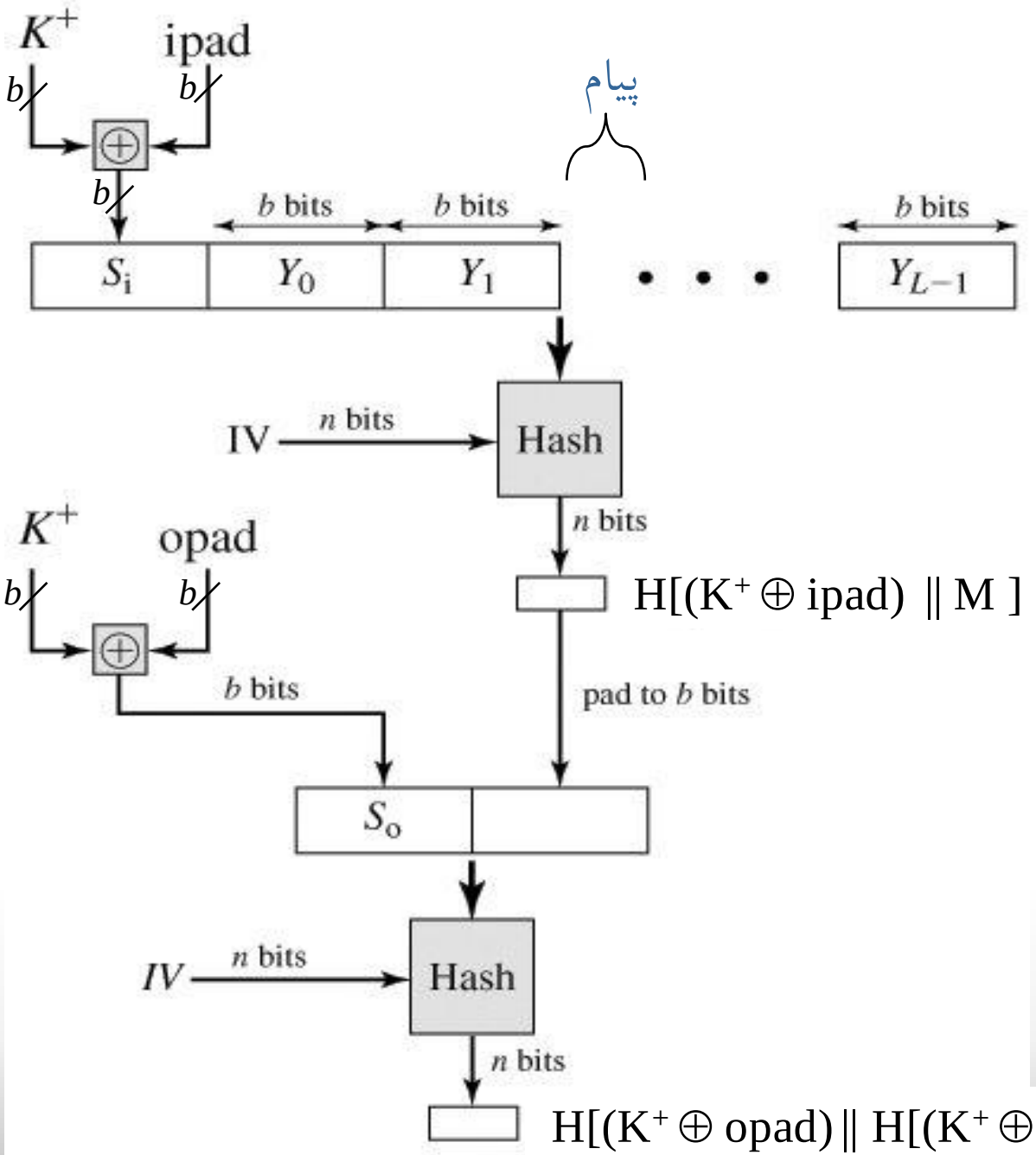
اهداف طراحی HMAC

- استفاده از توابع درهم‌ساز بدون تغییر آنها
- پشتیبانی از توابع درهم‌ساز متنوع
  - مانند MD5، SHA-1، SHA-2، Whirlpool و RIPEMD-160
- حفظ کارایی و سرعت تابع درهم‌ساز به کار گرفته شده
- استفاده ساده از کلید
- طراحی روشن و بدون ابهام

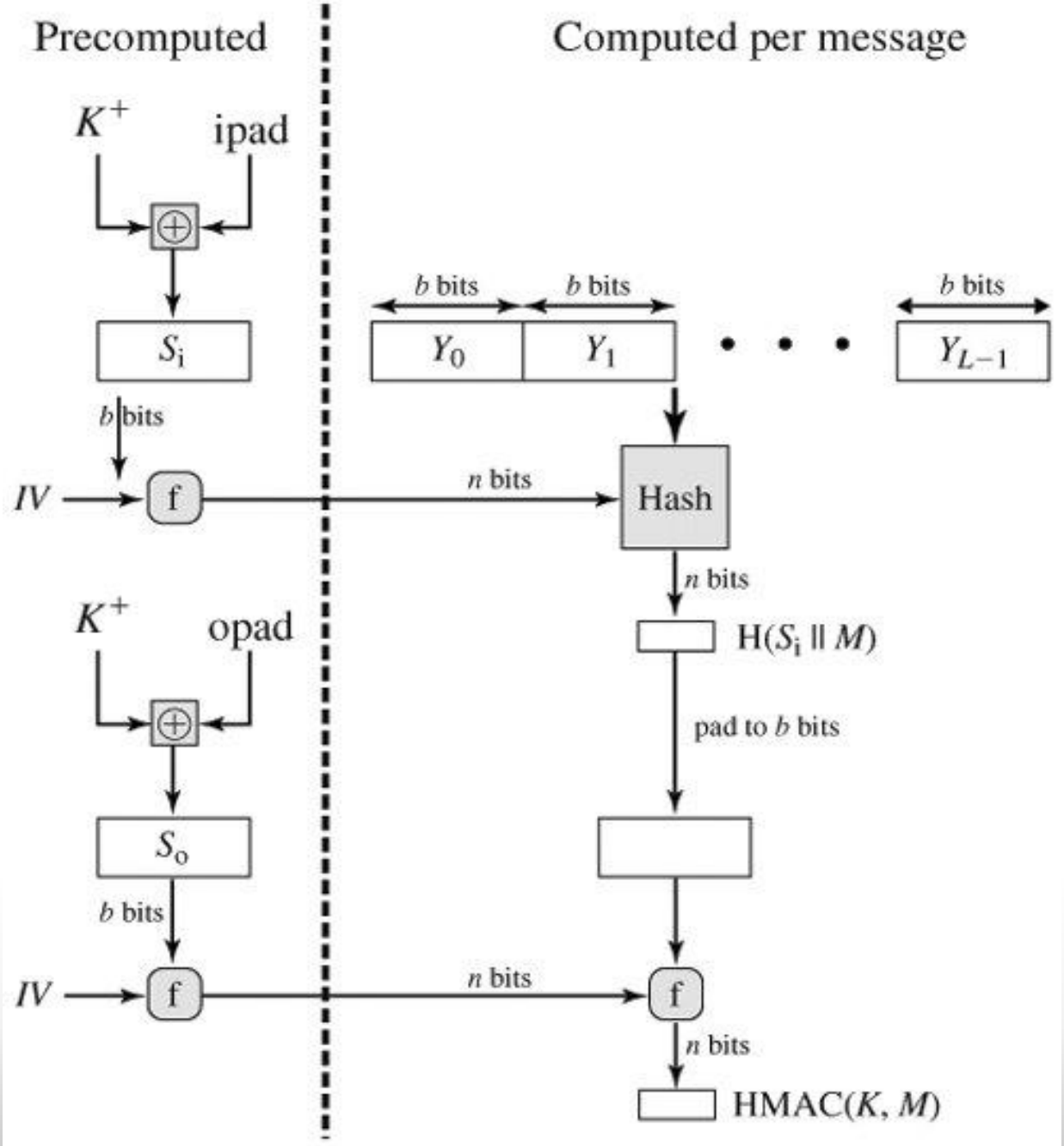
الگوریتم HMAC

- H: تابع درهم ساز 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 درهم‌ساز)
- K^+: کلید مخفی که یک دنباله صفر به سمت چپ آن اضافه شده است (تا به طول b برسد)
- ipad : رشته b بیتی حاصل از تکرار رشته ۰۰۱۱۰۱۱۰ به تعداد b/8
- opad: رشته b بیتی حاصل از تکرار رشته ۰۱۰۱۱۰۱۰ به تعداد b/8

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



نحوه محاسبه HMAC



پیاده‌سازی کارای 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 درهم‌ساز به کار گرفته شده، بیشتر است.
  - لذا استفاده از MD5 در هنگام نیاز به سرعت بیشتر مجاز است.

منابع

- اسلایدهای دکتر مرتضی امینی (منبع اصلی) – درس امنیت داده و شبکه