به نام خدا

دانشکده مهندسی برق،
دانشگاه صنعتی شریف

مبانی رمزنگاری و امنیت شبکه

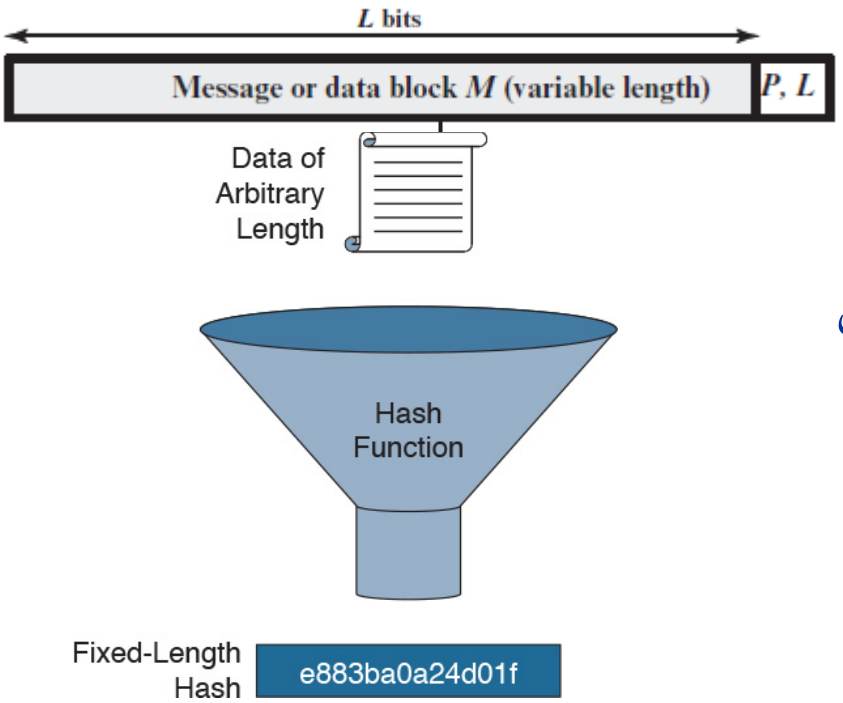
الگوریتم‌های یکپارچگی داده
توابع چکیده‌ساز، کدهای احراز اصالت پیام و امضای دیجیتال

Cryptographic Data Integrity Algorithms
Hash Functions, Message Authentication Codes and Digital Signature

مهتاب میرمحسنی

نیم‌سال دوم (بهار) ۹۸-۹۹

توابع چکیده‌ساز
Hash Functions



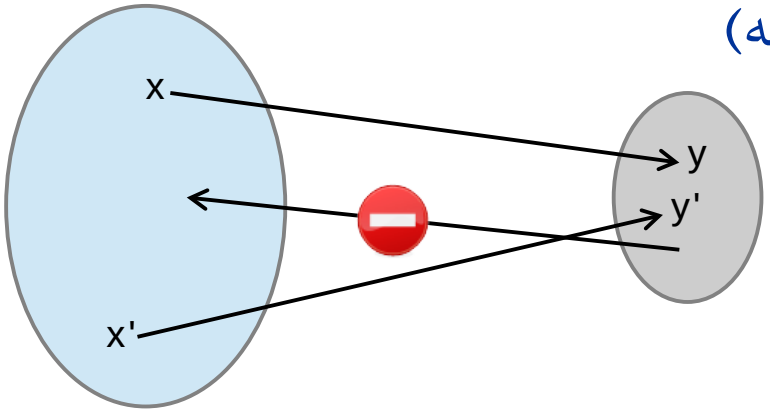
- تابع یکطرفه به طول ثابت
  - طول ورودی: متغیر (نامحدود)
  - طول خروجی: ثابت
- اگر به مجموعه بزرگی از ورودی‌ها اعمال شود، خروجی توزیع تقریبا یکنواخت داشته و تصادفی به نظر برسد
- اصل لانه کبوتری (تصادم- collision)

$$h = H(M)$$

$$x \neq y \;,\; H(x) = H(y)$$

توابع چکیده‌ساز
Cryptographic hash function

- محاسبه مستقیم ساده است
- محاسبه معکوس سخت است (یکطرفه)

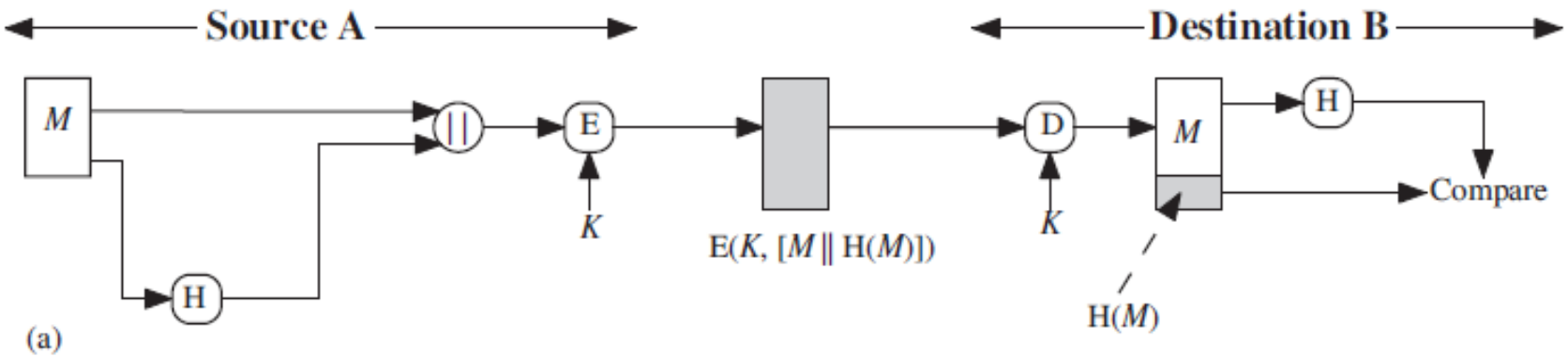


- خاصیت یک‌طرفه (one-way property)
  - یافتن پیام M متناظر با مقدار چکیده $h = H(M)$ از نظر محاسباتی غیر ممکن باشد
- خاصیت بدون برخورد (collision-free property)
  - یافتن دو پیام با مقدار چکیده یکسان، از نظر محاسباتی غیر ممکن باشد

کاربردهای تابع چکیده‌ساز
احراز اصالت پیام (Message Authentication)

- یکپارچگی پیام (بدون تغییر، درج، حذف یا تکرار)
- مقدار تابع چکیده‌ساز: چکیده پیام (message digest)

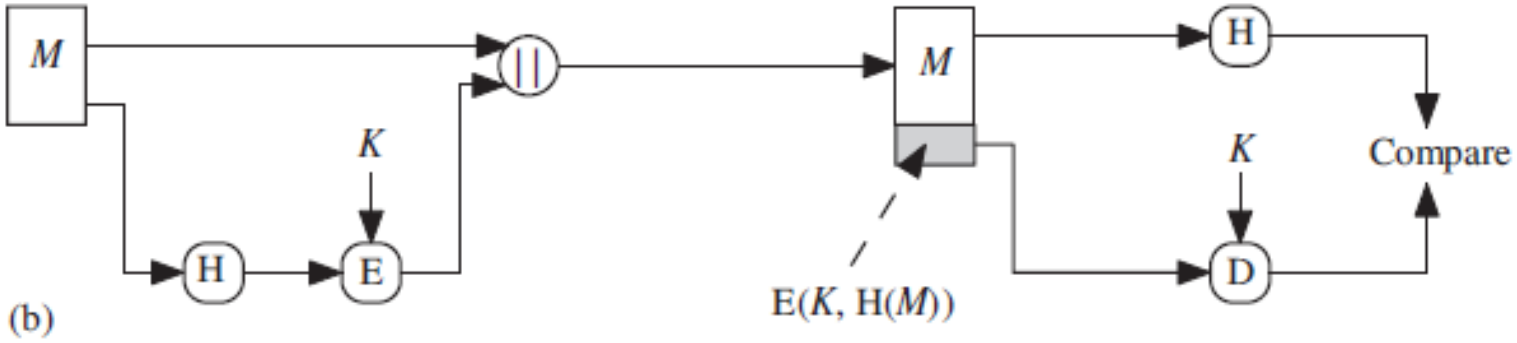
- **ترکیب با رمز متقارن**



- کلید مخفی: پیام از جانب A و یکپارچه است
- محرمانگی

کاربردهای تابع چکیده‌ساز
احراز اصالت پیام (Message Authentication)

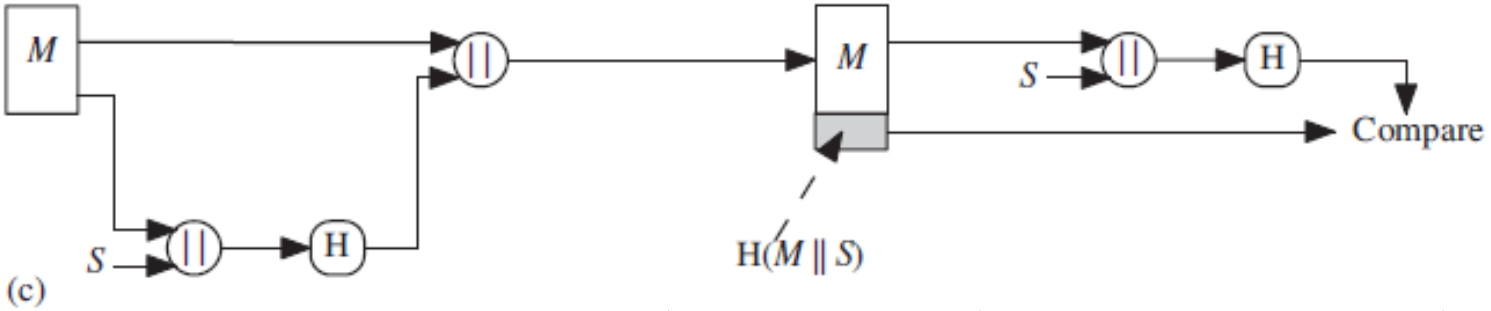
- تنها مقدار تابع چکیده‌ساز رمز شود (محرمانگی مورد نظر نیست)



- کد احراز اصالت پیام (message authentication code (MAC))
  - تابع MAC با ورودی کلید مخفی و قالب داده، چکیده‌ای را به عنوان کد می‌سازد
  - همراه پیام ذخیره و یا ارسال می‌شود
  - جهت احراز اصالت پیام، تابع MAC به پیام اعمال و با مقدار ذخیره شده مقایسه می‌شود
  - بدون آگاهی از کلید مخفی، تغییر دلخواه MAC ممکن نیست
  - با توجه به استفاده از کلید مخفی، فرستنده نیز احراز اصالت می‌شود
  - در عمل، الگوریتم‌های تولید MAC از رمزنگاری (E) کارآتر هستند

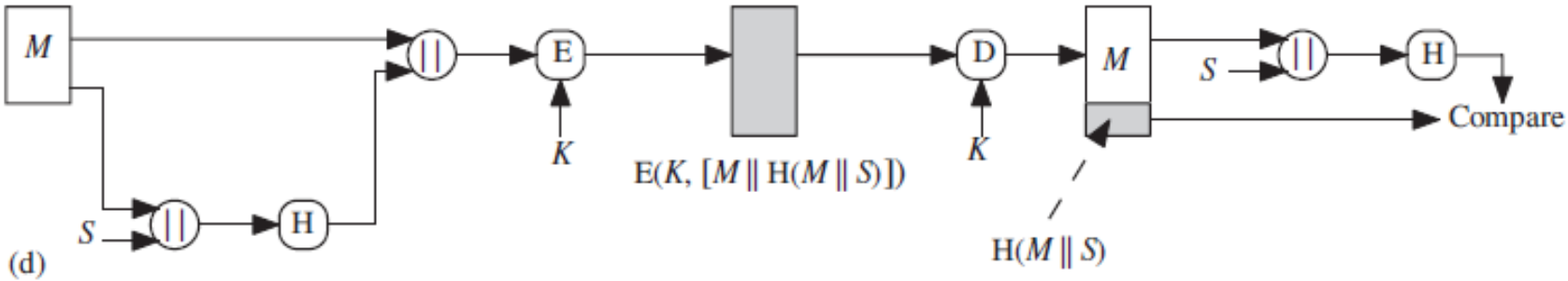
کاربردهای تابع چکیده‌ساز
احراز اصالت پیام (Message Authentication)

- بدون استفاده از رمزگذاری: تسهیم راز (secret sharing)



- چون راز ارسال نمی‌شود، دشمن قادر به تغییر پیام نیست

- محرمانگی

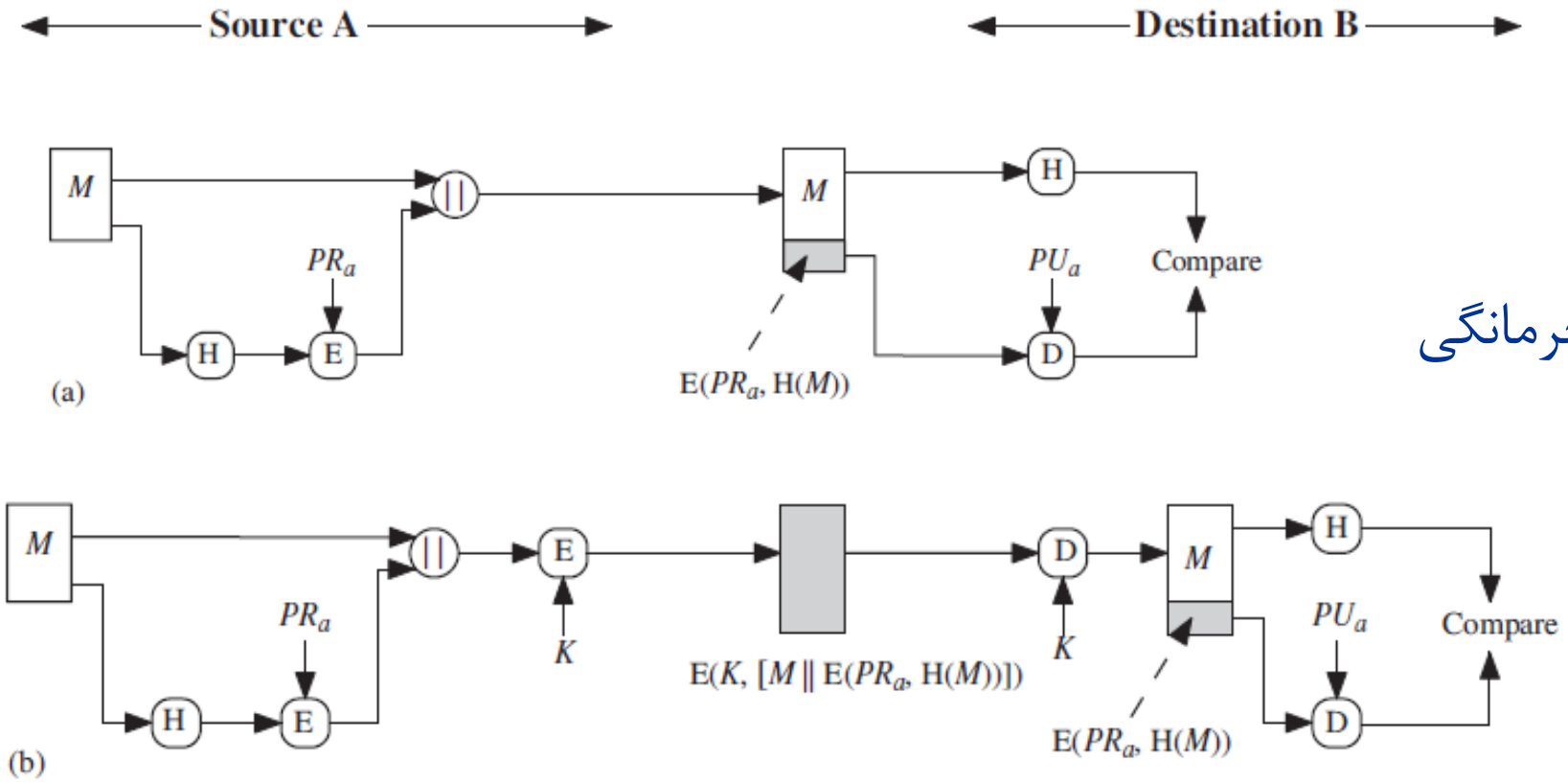


کاربردهای تابع چکیده‌ساز
احراز اصالت پیام (Message Authentication)

- اجتناب از رمزگذاری
- در بسیاری از کاربردها احراز اصالت پیام بدون محرمانگی ارجحیت دارد
- سرعت پایین رمزگذاری
  - سیستم‌هایی که همواره دنباله‌ای از پیام برای ارسال دارند
- هزینه بالای تجهیزات رمزگذاری
- الگوریتم‌های رمزنگاری برای داده‌هایی با طول زیاد بهینه هستند
  - برای طول کم، زمان زیادی صرف فرآیندهای مقداردهی اولیه می‌شود
- هزینه بدست آوردن امتیاز انحصاری الگوریتم‌های رمزنگاری

کاربردهای تابع چکیده‌ساز
امضای دیجیتال (Digital Signatures)

- با استفاده از کلید خصوصی فرستنده (رمزنگاری کلید همگانی یا نامتقارن)
- کاربرد آن فراتر از تنها احراز اصالت پیام است



- محرمانگی

کاربردهای دیگر تابع چکیده‌ساز

- فایل گذرواژه یک‌طرفه (one-way password file)
  - چکیده گذرواژه ذخیره می‌گردد
  - پس از وارد شدن گذرواژه، چکیده آن محاسبه و با مقدار ذخیره شده مقایسه می‌شود
  - رخنه‌گر (hacker) دسترسی به گذرواژه ندارد
  - در بیشتر سیستم عامل‌ها به کار می‌رود
- تشخیص نفوذ (intrusion detection) و تشخیص ویروس (virus detection)
  - برای هر فایل در سیستم H(F) محاسبه و به صورت امن ذخیره می‌شود
  - با محاسبه مجدد H(F) در هر زمان، تغییرات آشکار می‌شود
  - نفوذگر (intruder) بدون تغییر H(F) نمی‌تواند فایل را تغییر دهد
- تولید اعداد شبه تصادفی (pseudorandom number generator (PRNG))
  - تولید کلید در رمزنگاری متقارن (hash-based PRF)

توابع چکیده‌ساز ساده

- ناامن (insecure)
- چکیده n بیتی
- ورودی به m قالب n بیتی تقسیم می‌شود
- ساده‌ترین: XOR بیتی قالب‌ها

$$C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$$

- داده تصادفی: احتمال عدم تغییر چکیده با تغییر داده $= 2^{-n}$
- در متن ساختار یافته اردر n کم می‌شود
- بهبود:
  - هر قالب مرحله به مرحله XOR شده و در مقدار چکیده قرار می‌گیرد
  - پیش از XOR کردن قالب جدید، مقدار چکیده ۱ بیت شیفت داده می‌شود

توابع چکیده‌ساز

- x پیش‌تصویر h است اگر $h = H(x)$
- H: نگاشت چند-به-یک ← برای هر مقدار h، چندین پیش‌تصویر وجود دارد

- برخورد (collision) زمانی رخ می‌دهد که:

$$x \neq y \;,\; H(x) = H(y)$$

  - کاربرد در یکپارچگی داده ← برخورد نامطلوب است
  - حتما برخورد وجود دارد ← یافتن آن باید از نظر محاسباتی غیرممکن باشد
  - طول ورودی $= b$، طول چکیده $= n$، به طور متوسط 2^{b-n} برخورد

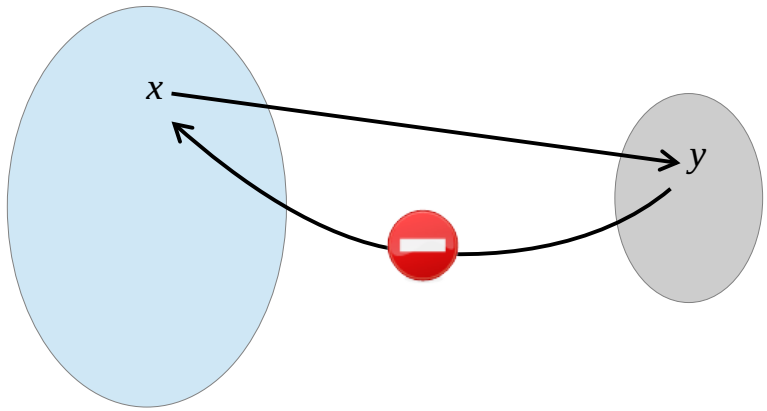
الزامات امنیتی توابع چکیده‌ساز

الزامات پیاده‌سازی عملی

- طول ورودی متغیر
  - H را می‌توان به آرگومانی با هر اندازه دلخواه اعمال کرد (بعد دامنه بی‌نهایت است)
- طول خروجی ثابت
  - H یک خروجی با اندازه ثابت دارد (بعد برد، عدد ثابت n است)
- کارآیی: محاسبه $H(x)$ برای x دلخواه به سادگی انجام می‌شود

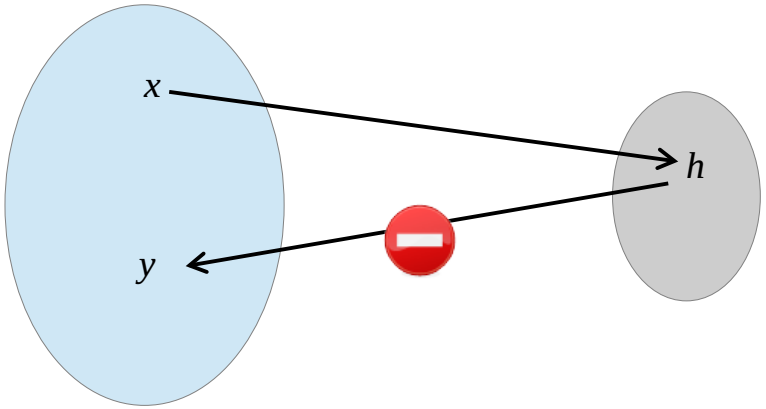
الزامات امنیتی توابع چکیده‌ساز

- مقاوم در برابر پیش‌تصویر: خاصیت یک‌طرفه
  - Preimage resistance (one-way property)
  - برای هر y داده شده، بدست آوردن x به طوری‌که $y=H(x)$ باشد، از نظر محاسباتی غیرممکن است



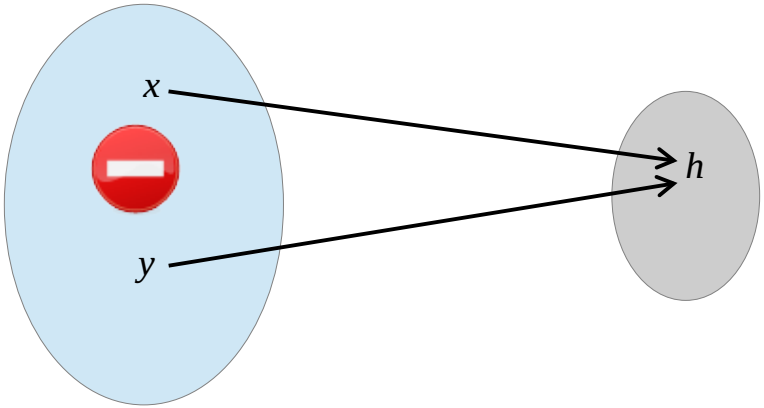
الزامات امنیتی توابع چکیده‌ساز

- مقاوم در برابر برخورد ضعیف (مقاوم در برابر پیش‌تصویر دوم)
  - Second-preimage resistance (weak collision resistance)
  - برای یک x داده شده، بدست آوردن $y \neq x$ به طوری‌که $H(y)=H(x)$ باشد، از نظر محاسباتی غیرممکن است



الزامات امنیتی توابع چکیده‌ساز

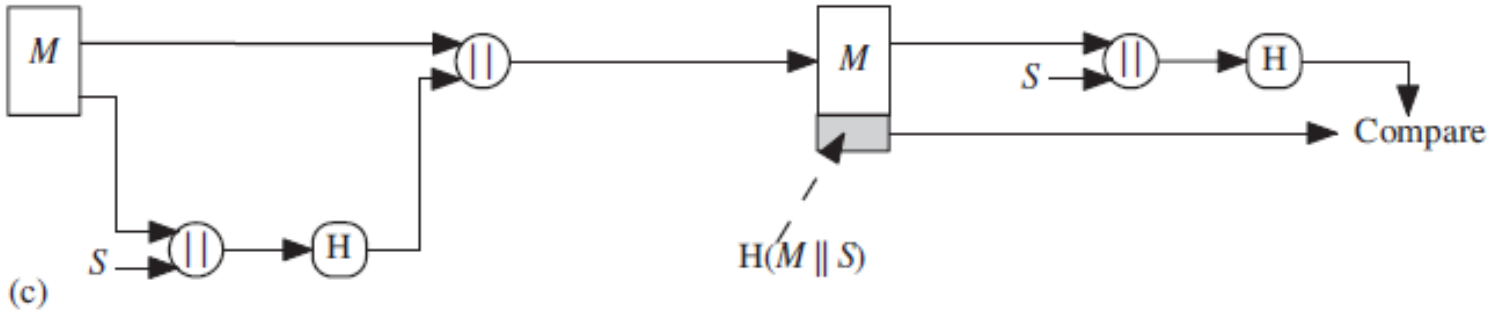
- مقاوم در برابر برخورد قوی (مقاوم در برابر برخورد)
  - Collision resistance (strong collision resistance)
  - بدست آوردن x و y، به طوری‌که $H(y)=H(x)$ باشد، از نظر محاسباتی غیرممکن است



- شبه تصادفی بودن
  - خروجی در تست‌های آماری صدق کند
  - الزامی نیست ولی معمول است

مقاوم در برابر پیش‌تصویر (preimage resistant)
خاصیت یک‌طرفه

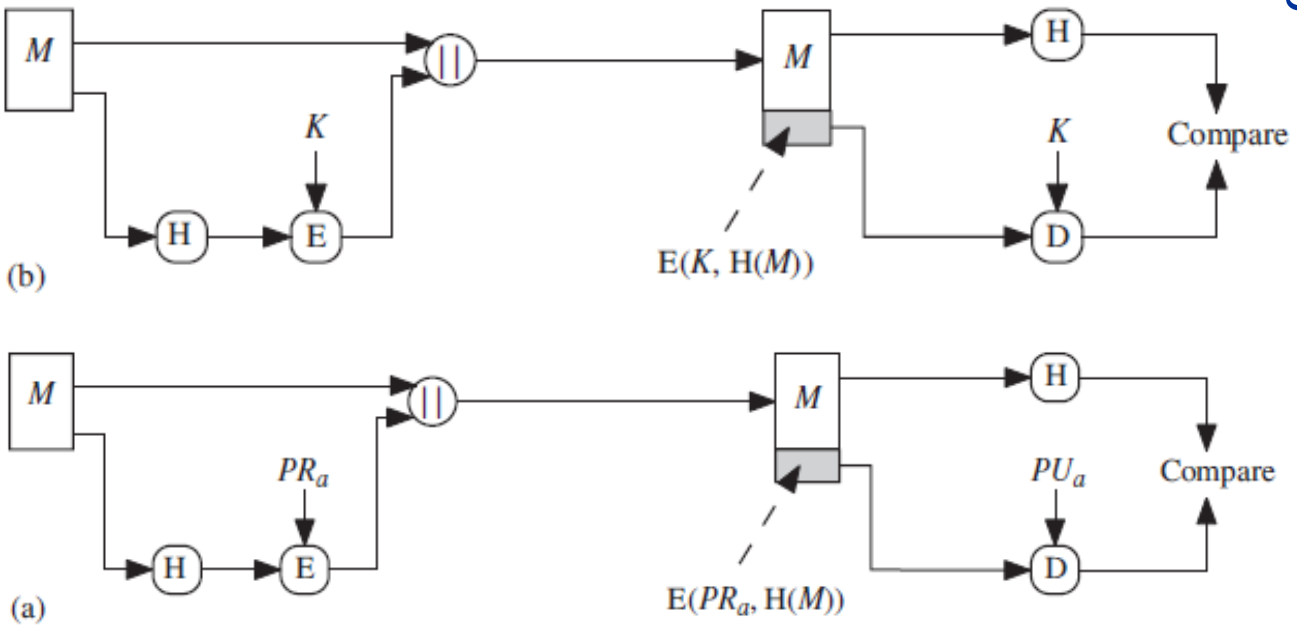
- تولید کد (چکیده) برای یک پیام ساده است ولی بدست آوردن پیام با داشتن کد از نظر محاسباتی غیرممکن است
- احراز اصالت با بکارگیری راز



- دشمن راز را بازیابی می‌کند!

مقاوم در برابر برخورد ضعیف (مقاوم در برابر پیش‌تصویر دوم)

- پیداکردن پیام دیگری با چکیده برابر با چکیده پیام اصلی از نظر محاسباتی غیرممکن است
- کد چکیده رمزگذاری شده
- مهاجم پیام و چکیده رمزگذاری شده را می‌بیند
- چکیده پیام را می‌سازد
- پیام دلخواهی را می‌یابد که چکیده آن برابر با چکیده پیام اصلی باشد



مقاوم در برابر برخورد قوی (مقاوم در برابر برخورد)

حمله:

- آلیس مبلغ کمی به باب قرض دارد و قرار است پیام حاوی اعلام این قرض را امضا کند
- باب دو پیام که دارای چکیده یکسان هستند را می‌یابد که یکی حاوی مبلغ کم قرض و دیگری حاوی مبلغ زیاد قرض است
- پیام اصلی (که حاوی مبلغ کم قرض است) را برای آلیس می‌فرستد. آلیس آن را امضا می‌کند و پس می‌فرستد
- باب چکیده امضا شده را در کنار پیام دوم قرار می‌دهد و ادعای اصالت آن را می‌کند!!

الزامات توابع چکیده‌ساز در کاربردهای مختلف

	Preimage Resistant	Second Preimage Resistant	Collision Resistant
Hash + digital signature	yes	yes	yes*
Intrusion detection and virus detection		yes	
Hash + symmetric encryption			
One-way password file	yes		
MAC	yes	yes	yes*

* Resistance required if attacker is able to mount a chosen message attack

حملات جستجوی فراگیر

- به تعداد بیت مقدار چکیده (اندازه خروجی) بستگی دارد
- **حملات پیش‌تصویر و پیش‌تصویر دوم**
- در حمله پیش‌تصویر، دشمن به دنبال پیام y است، به طوریکه چکیده آن ($\mathrm{H}(y)$) مقدار مشخص h باشد
- حمله جستجوی فراگیر: انتخاب y تصادفی و تکرار تا یافتن برخورد
- برای مقدار چکیده m بیتی
  - دشمن باید به طور متوسط 2^{m-1} تعداد y را بیازماید تا به مقدار چکیده دلخواه برسد

حملات جستجوی فراگیر

- سوال: تابع H با 2^m خروجی ممکن را در نظر بگیرید (m بیتی). مقدار مشخص $H(x)$ داده شده است. اگر H به k مقدار تصادفی اعمال شود، k چقدر باید باشد تا احتمال این که حداقل یک y پیدا شود که $H(y)=H(x)$ است، بزرگ‌تر از ۰.۵ باشد؟

$$k = 2^{m-1}$$

پارادوکس روز تولد

- حداقل مقدار k چقدر باید باشد، تا در یک گروه k نفری احتمال این که حداقل دو نفر در یک روز متولد شده باشند، بیشتر از ۰.۵ باشد؟

$k = 23$

- تابع H با 2^m خروجی ممکن را در نظر بگیرید (m بیتی).
- H را به k مقدار ورودی تصادفی اعمال کرده و نتیجه را در مجموعه X ذخیره می‌کنیم
- دوباره، H را به k مقدار ورودی تصادفی اعمال کرده و نتیجه را در مجموعه Y ذخیره می‌کنیم
- k چقدر باید باشد تا احتمال این که حداقل یک تطابق در دو مجموعه رخ دهد، بزرگ‌تر از ۰.۵ باشد؟

$H(x) = H(y)$ for some $x \in X, y \in Y$ $\qquad k = \sqrt{2^m} = 2^{m/2}$

- متغیر تصادفی با توزیع یکنواخت بین 0 و N-1: پس از انتخاب $\sqrt{N}$ تعداد از آن با احتمال بیشتر از ۰.۵، یک مقدار تکراری انتخاب می‌شود

حملات جستجوی فراگیر
حمله مقاوم در برابر برخورد

- مهاجم به دنبال یافتن پیام‌های x و y ای است که چکیده یکسان داشته باشند: $H(x)=H(y)$
- چکیده m بیتی: $\sqrt{2^m}=2^{m/2}$

استفاده از پارادوکس روز تولد در حمله مقاوم در برابر برخورد (امضای دیجیتال)

- منبع A قصد امضای پیام قانونی x با کلید خصوصی خود را دارد
- مهاجم، $2^{m/2}$ پیام دگرگون شده از x را که اساساً هم معنا هستند تولید می‌کند (x')
- مهاجم پیام تقلبی دلخواه خود (y) را که نیاز به امضای A دارد، تولید می‌کند
- مهاجم، پیام‌های دگرگون شده از y را که اساساً هم معنا هستند تولید می‌کند (y')
  - مقدار چکیده آن را تولید و با مقادیر چکیده‌های قبلی مقایسه می‌کند $H(x')\stackrel{?}{=}H(y')$
- مهاجم، پیام قانونی را برای امضا به A می‌فرستد و امضای دریافتی را به پیام تقلبی الصاق می‌کند

64-bit hash $\rightarrow 2^{32}$

Dear Anthony,

{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]

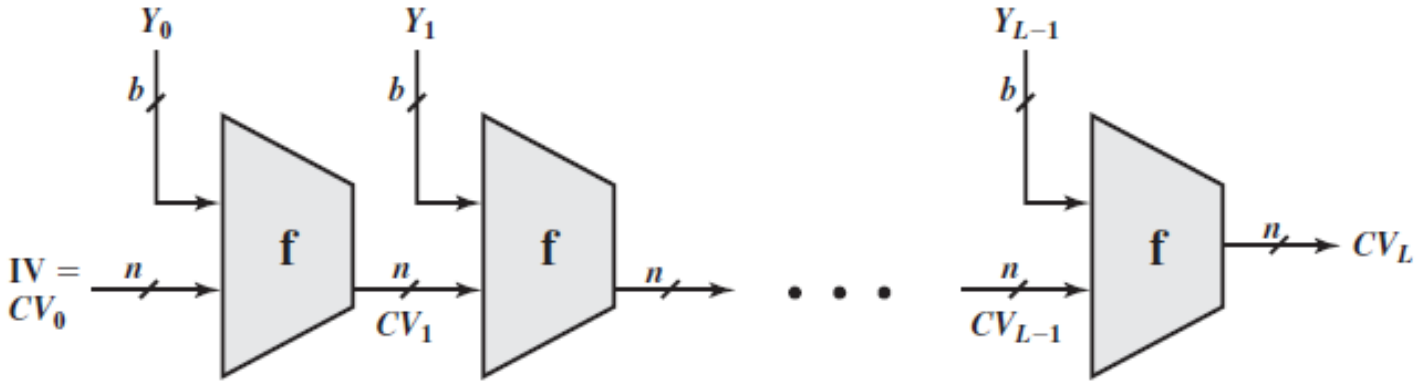
امنیت توابع چکیده‌ساز

Preimage resistant	2^m
Second preimage resistant	2^m
Collision resistant	$2^{m/2}$

- MD5: ۱۲۸ بیت
  - ۱۹۹۴ – ۱۰ میلیون دلار – ۲۴ روز
- ۱۶۰ بیت
  - بیش از ۴۰۰۰ سال
- فناوری امروزه: ۱۶۰ بیت مشکوک است!

ساختار کلی توابع چکیده‌ساز

- مرکل (۱۹۷۹): تابع چکیده‌ساز مکرر
  - مبنای اکثر توابع چکیده‌ساز مانند SHA
- پیام به L قالب b بیتی تقسیم می‌شود
- اعمال مکرر تابع فشرده‌ساز f
- اگر f در برابر برخورد مقاوم باشد، تابع چکیده‌ساز مکرر نیز مقاوم است



IV = Initial value
CV_i = 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

الگوریتم چکیده‌ساز امن
Secure Hash Algorithm (SHA)

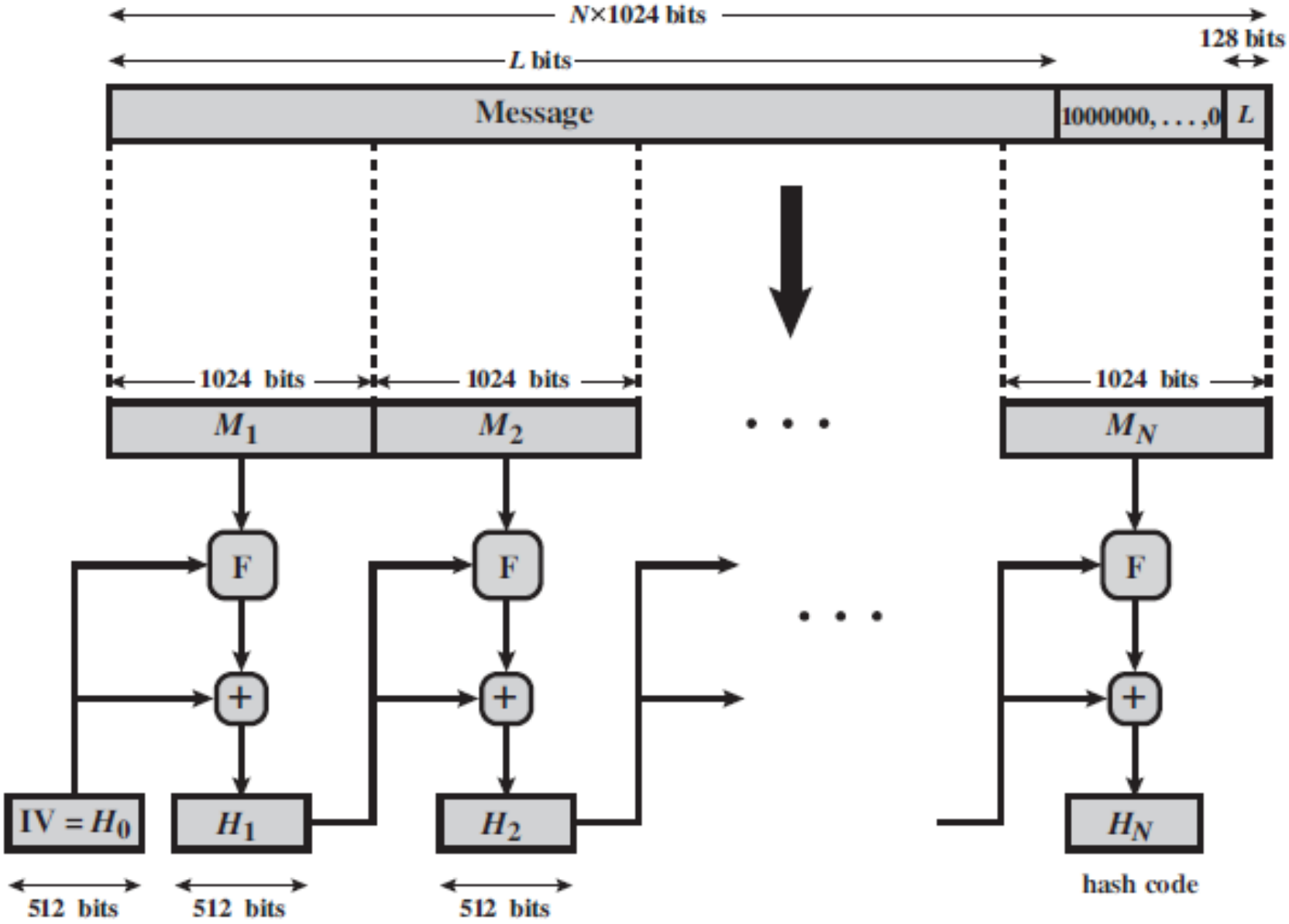
- پرکاربردترین توابع چکیده‌ساز
  - از ۲۰۰۵: تنها الگوریتمی که در برابر حملات رمزشکنی شکسته نشده‌اند
- استاندارد NIST در ۱۹۹۳ (FIPS 180)
  - SHA-0
- SHA-1 (۱۹۹۵)
  - ۱۶۰ بیت
  - استاندارد امضای دیجیتال DSS (ELGAMAL)
  - در ۲۰۰۵ پایان مقبولیت استفاده از آن تا ۲۰۱۰ اعلام شد
  - حمله‌ای پیدا شد که تعداد عمل برای یافتن برخورد را از 2^{80} به 2^{69} کاهش داد

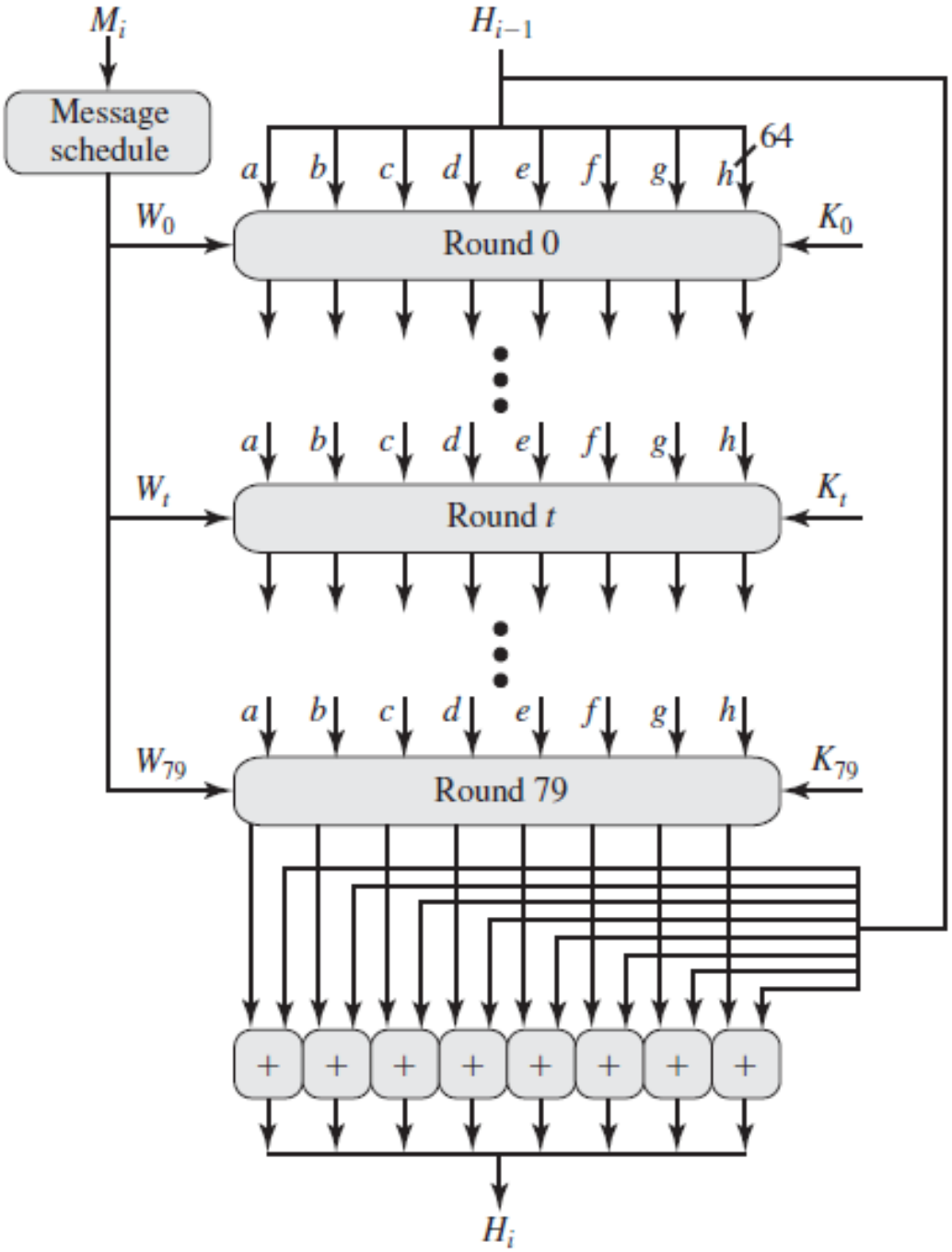
الگوریتم چکیده‌ساز امن
Secure Hash Algorithm (SHA)

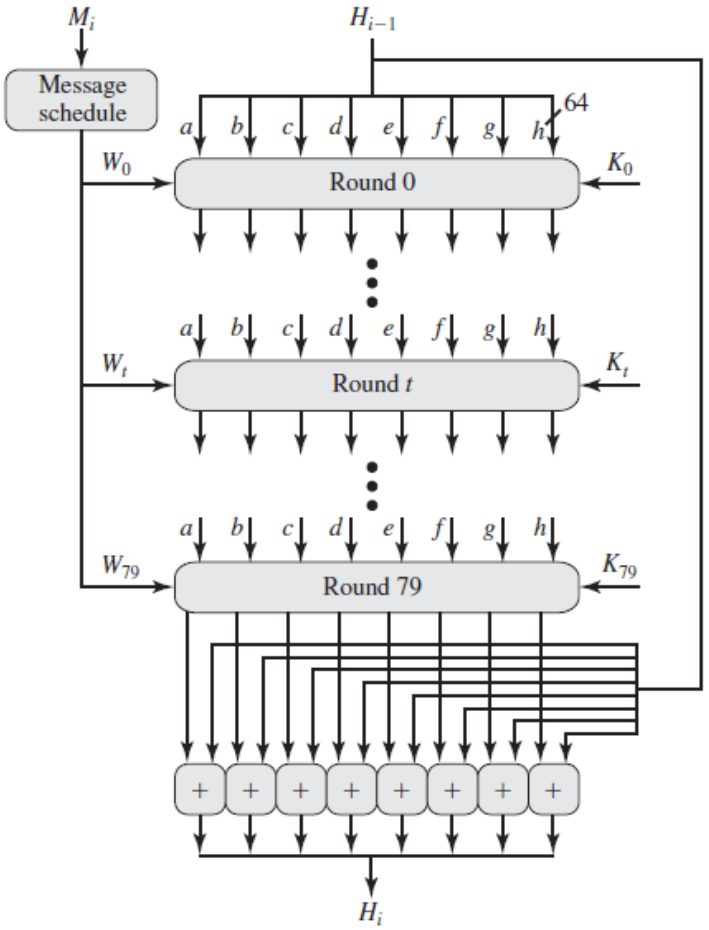
- SHA-2 (۲۰۰۲)
  - ۲۵۶، ۳۸۴ و ۵۱۲ بیت
  - مبنای ساختاری یکسان با SHA-1
  - ۲۲۴ بیت (۲۰۰۸)

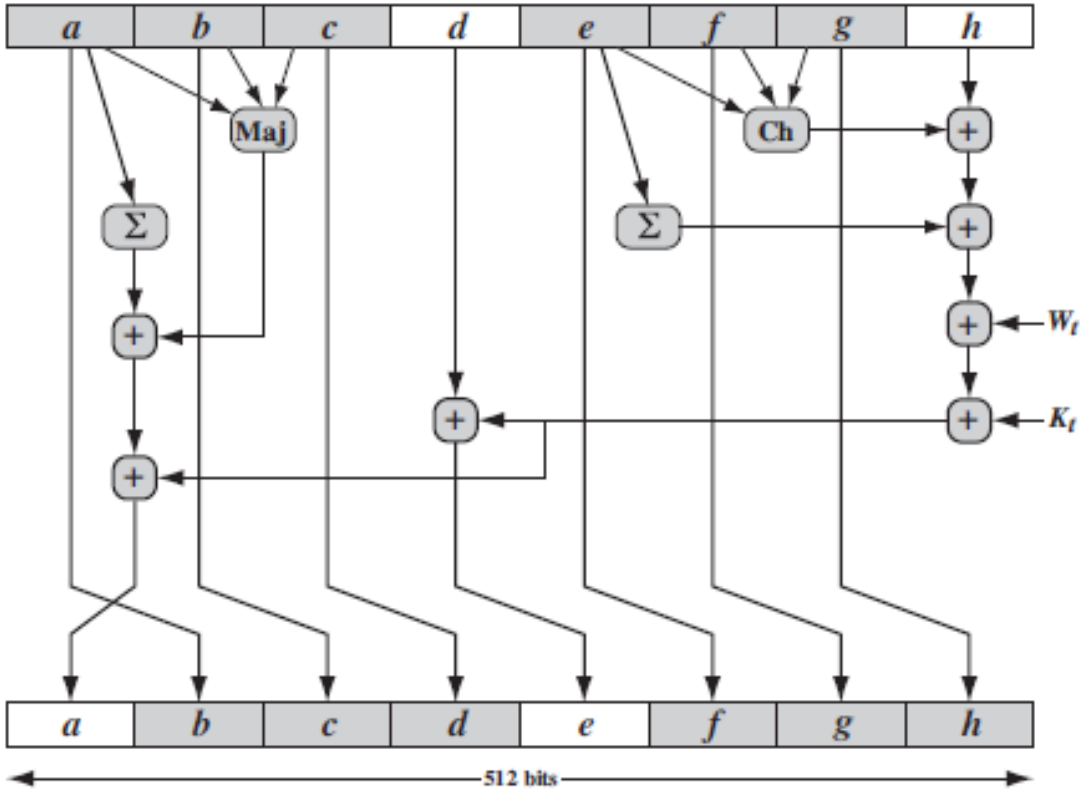
Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

SHA-512


N×1024 bits
128 bits
L bits
Message
1000000, . . . ,0
L
1024 bits
1024 bits
1024 bits
M1
M2
MN
F
F
F
+
+
+
IV = H0
H1
H2
HN
512 bits
512 bits
512 bits
hash code


M_i
H_{i-1}
Message schedule
64
a b c d e f g h
W_0
Round 0
K_0
a b c d e f g h
W_t
Round t
K_t
a b c d e f g h
W_{79}
Round 79
K_{79}
+ + + + + + + +
H_i





$$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}$$

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

SHA-3

- SHA-2
  - مبنای ساختاری یکسان با استانداردهای قبلی
- مسابقه‌ای جهت انتخاب استاندارد جدید (۲۰۰۷)
- امکان جایگزینی SHA-2 با SHA-3 در کاربردهای فعلی باشد
  - ۲۲۴، ۲۵۶، ۳۸۴ و ۵۱۲ بیت را بتواند تولید کند
- خاصیت برخط (online) موجود در SHA-2 را حفظ کند
  - بر روی قالب‌های کوتاه (مثل ۵۱۲ یا ۱۰۲۴ بیت) عمل کرده و منتظر تمام پیام نباشد
- امنیت – هزینه – پیاده‌سازی
- در ۲ اکتبر ۲۰۱۲ برنده مسابقه SHA-3 اعلام شد
  - **Keccak** با تیم طراحانی از بلژیک و ایتالیا
  - استاندارد در ۲۰۱۵

حملات امنیتی

- افشا (Disclosure)
- تحلیل ترافیک
- رخ‌پوشی (masquerade): وارد کردن پیام در شبکه توسط منبع غیرقانونی
  - تولید پیام غیرقانونی و ادعای اصالت آن
  - اعلام دریافت یا عدم دریافت غیرقانونی (توسط شخصی به جز گیرنده اصلی)
- تغییر محتوا (درج، حذف، جابجایی و یا تغییر)
- تغییر دنباله (درج، حذف و یا تغییر ترتیب)
- تغییر زمانی (تاخیر یا تکرار)
- انکار منبع (Source repudiation)
- انکار مقصد (Destination repudiation)

حملات امنیتی

- افشا (Disclosure)
- تحلیل ترافیک

محرمانگی

- رخ‌پوشی (masquerade): وارد کردن پیام در شبکه توسط منبع غیرقانونی
  - تولید پیام غیرقانونی و ادعای اصالت آن
  - اعلام دریافت یا عدم دریافت غیرقانونی (توسط شخصی به جز گیرنده اصلی)
- تغییر محتوا (درج، حذف، جابجایی و یا تغییر)
- تغییر دنباله (درج، حذف و یا تغییر ترتیب)
- تغییر زمانی (تاخیر یا تکرار)
- انکار منبع (Source repudiation)
- انکار مقصد (Destination repudiation)

احراز اصالت پیام

امضای دیجیتال

امضای دیجیتال+پروتکل

توابع احراز اصالت پیام

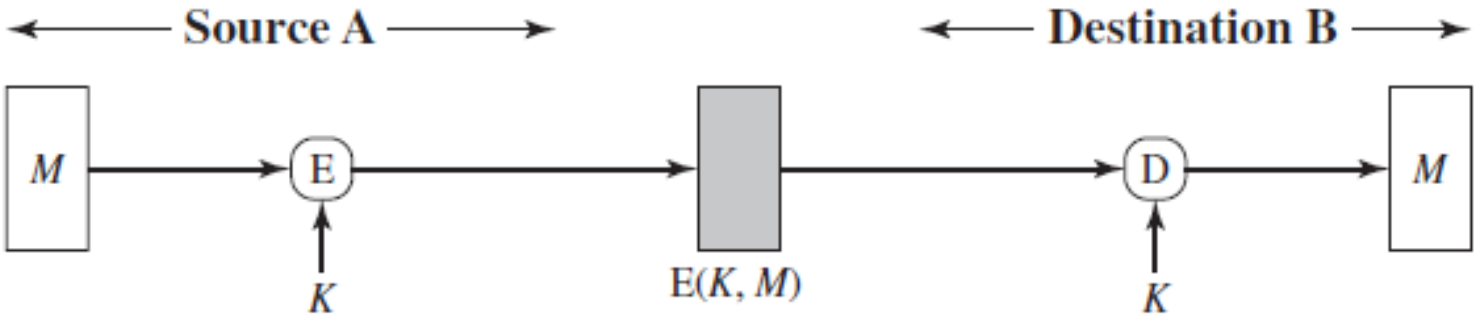
- ساز و کار احراز اصالت پیام یا امضای دیجیتال: ۲ مرحله
  1. تابعی که مقداری به نام احراز اصالت‌گر (authenticator) را تولید می‌کند
  2. پروتکل احراز اصالت که گیرنده با استفاده احراز اصالت‌گر، پیام را احراز اصالت کند
- توابع چکیده‌ساز (Hash function)
  - احراز اصالت‌گر = خروجی (طول ثابت) تابع چکیده‌ساز با ورودی پیام
- رمزنگاری پیام (Message encryption)
  - احراز اصالت‌گر = متن رمز شده کل پیام
- کد احراز اصالت پیام (Message authentication code (MAC))
  - احراز اصالت‌گر = خروجی (طول ثابت) یک تابع با ورودی پیام و **کلید مخفی**

رمزنگاری پیام (Message encryption)
رمز متقارن

- کلید مخفی: محرمانگی
- B می‌داند که پیام توسط A تولید شده است و تغییر نیافته است
  - احراز اصالت؟
- آیا Y متن اصلی قانونی است؟ روش خودکار

$$M \xrightarrow{E} X \rightarrow Y = D(K, X)$$

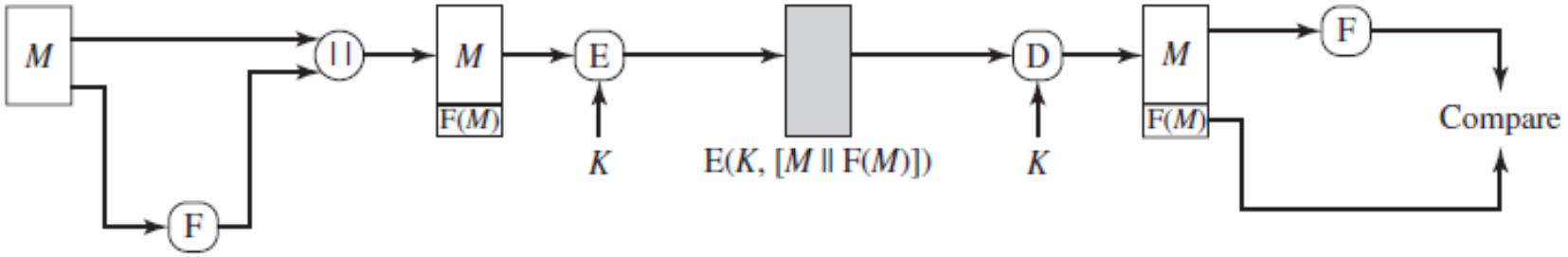
  - M: هر دنباله‌ای از بیت‌ها ← غیر قابل تشخیص ← Y می‌تواند هر دنباله‌ای از بیت‌ها باشد
- در حالت کلی تشخیص متن اصلی قانونی به صورت خودکار ساده نیست
  - راه حل: متن اصلی دارای ساختار باشد



(a) Symmetric encryption: confidentiality and authentication

رمزنگاری پیام (Message encryption)
رمز متقارن

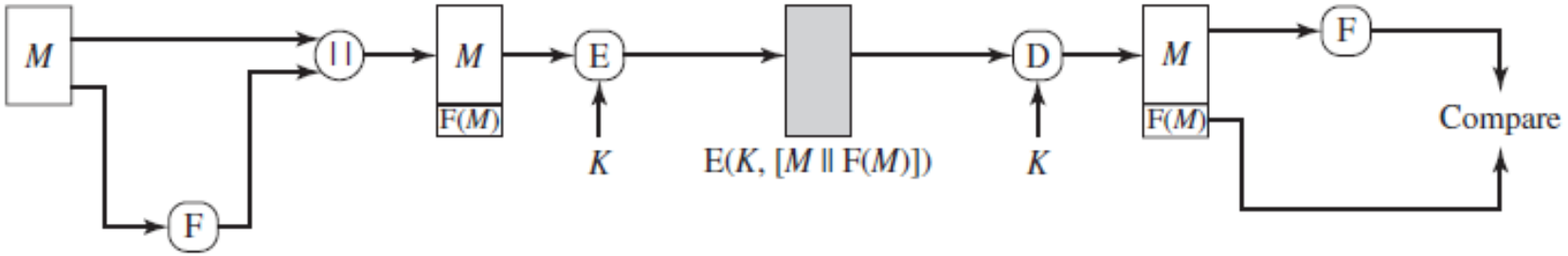
- اضافه کردن کد تشخیص خطا یا جمع آزما (checksum) به هر پیام:
  - کنترل خطای درونی: دشمن نمی‌تواند متن رمزشده‌ای تولید کند که پس از رمزگشایی با کد تشخیص خطا متناظر باشد



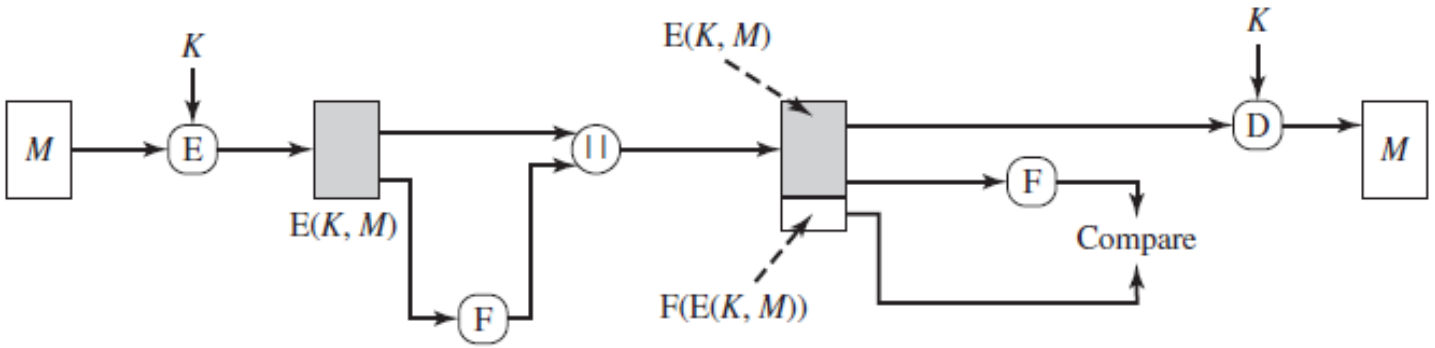
(a) Internal error control

رمزنگاری پیام (Message encryption)
رمز متقارن

- اضافه کردن کد تشخیص خطا یا جمع آزما (checksum) به هر پیام:
  - کنترل خطای درونی: دشمن نمی‌تواند متن رمزشده‌ای تولید کند که پس از رمزگشایی با کد تشخیص خطا متناظر باشد
  - ترتیب مهم (کنترل خطای بیرونی): دشمن می‌تواند متن رمزشده با کد تشخیص خطای متناظر بسازد، هرچند متن اصلی را نمی‌داند



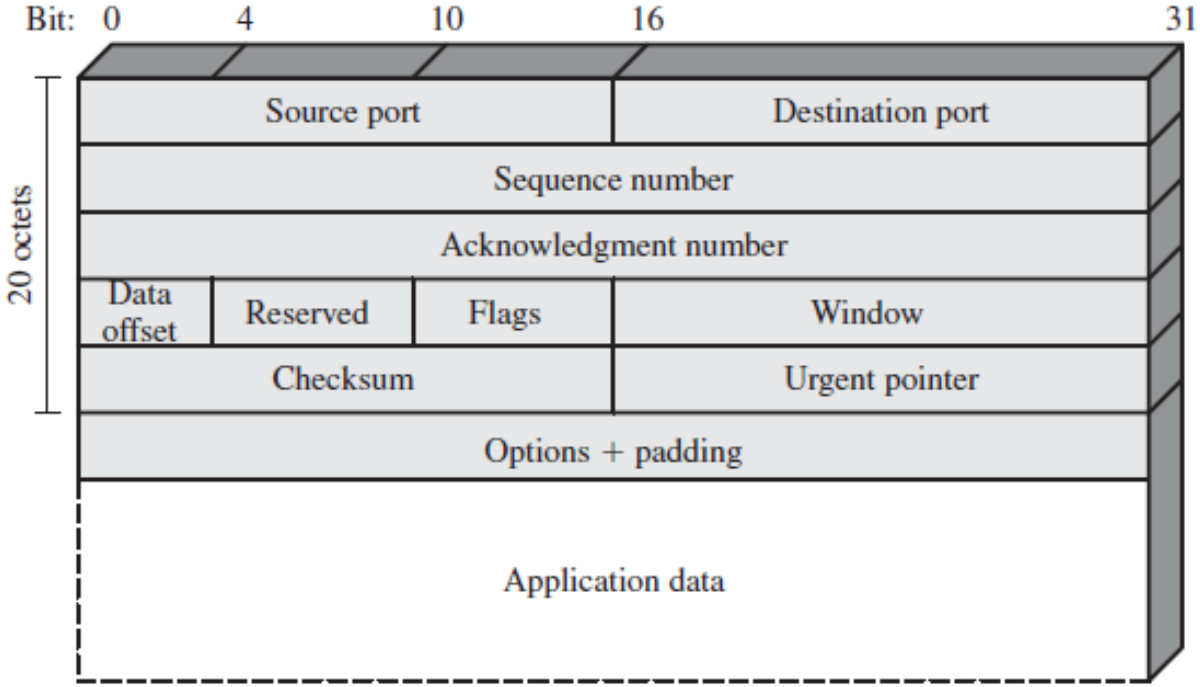
(a) Internal error control



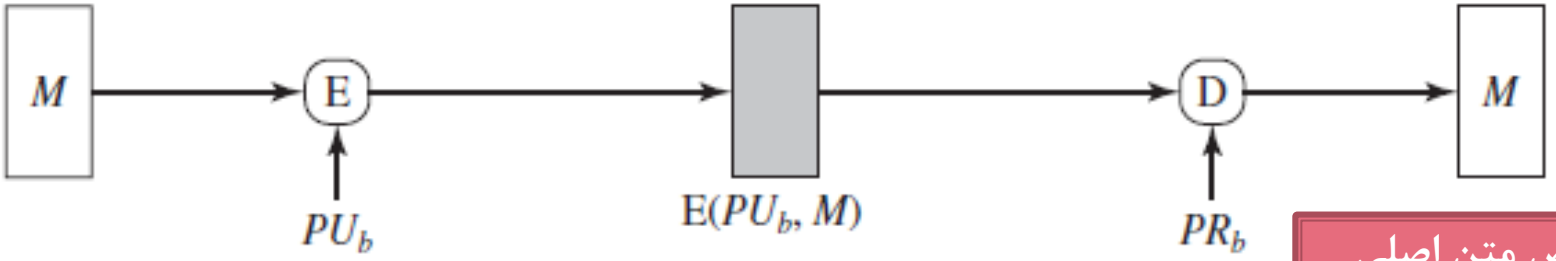
(b) External error control

رمزنگاری پیام (Message encryption)
رمز متقارن

- هر ساختار دیگر در پیام (مانند پروتکل‌های ارتباطی در لایه‌های مختلف شبکه)
- پروتکل TCP/IP
- کلید مخفی بین هر دو کاربر
- سرایند (header) شامل ساختار لازم و حتی جمع آزما است
- شماره دنباله
  - تاخیر، حذف و تغییر ترتیب

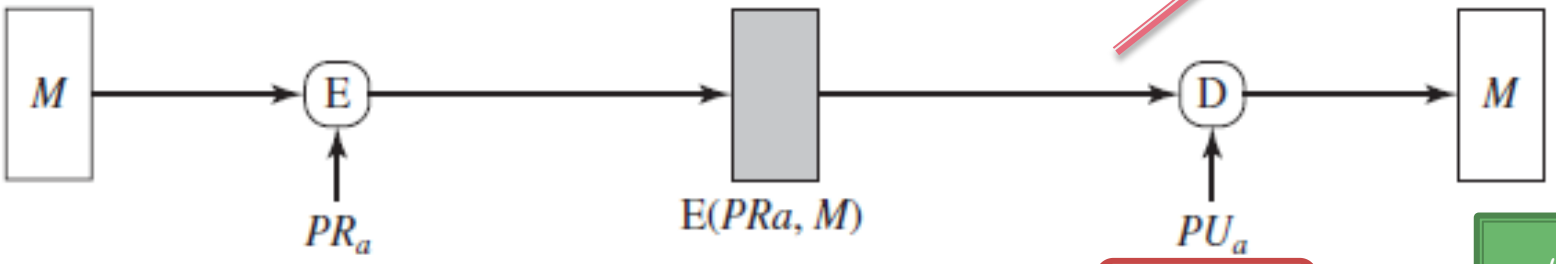


رمزنگاری پیام (Message encryption)
رمز نامتقارن (کلید همگانی)



(b) Public-key encryption: confidentiality

تشخیص متن اصلی
قانونی خودکار



(c) Public-key encryption: authentication and signature

۴ بار اجرای
الگوریتم پیچیده

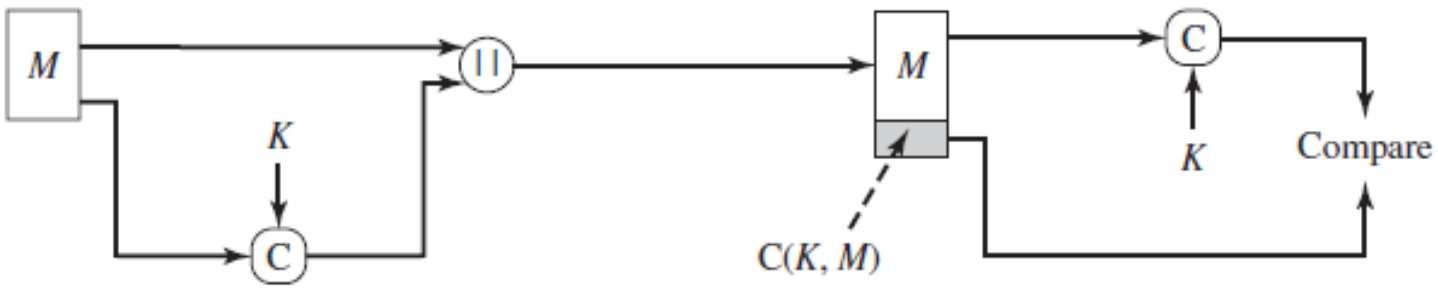
(d) Public-key encryption: confidentiality, authentication, and signature

کد احراز اصالت پیام
Message authentication code (MAC)

- کد احراز اصالت پیام یا جمع‌آزمای رمزنگاشتی (cryptographic checksum)
- طول خروجی: ثابت
- تابعی از پیام (M) و **کلید مخفی** (K)

$$\mathrm{MAC} = C(K, M)$$

- پیام همراه با مقدار MAC ارسال می‌شود
- گیرنده با استفاده از کلید مخفی، MAC را محاسبه و با مقدار دریافتی مقایسه می‌کند
- مهاجم بدون کلید مخفی نمی‌تواند MAC بسازد ← پیام تغییر نیافته است
- تنها فرستنده کلید مخفی را می‌داند و می‌تواند MAC بسازد ← احراز اصالت فرستنده M
- اگر پیام حاوی شماره دنباله باشد (مثل TPC) ← ترتیب تغییر نیافته است



(a) Message authentication

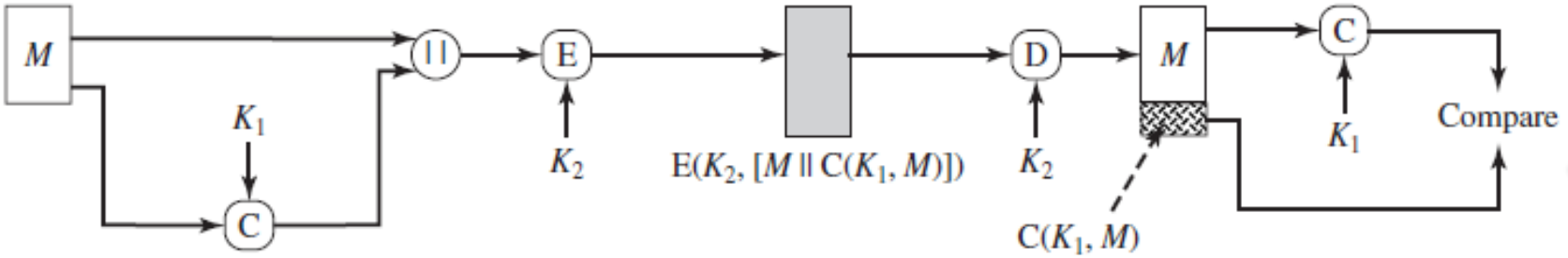
کد احراز اصالت پیام
Message authentication code (MAC)

- تابع MAC: برخلاف رمزنگاری نیازی به وارون‌پذیری نیست
  - وارون‌پذیر نیست

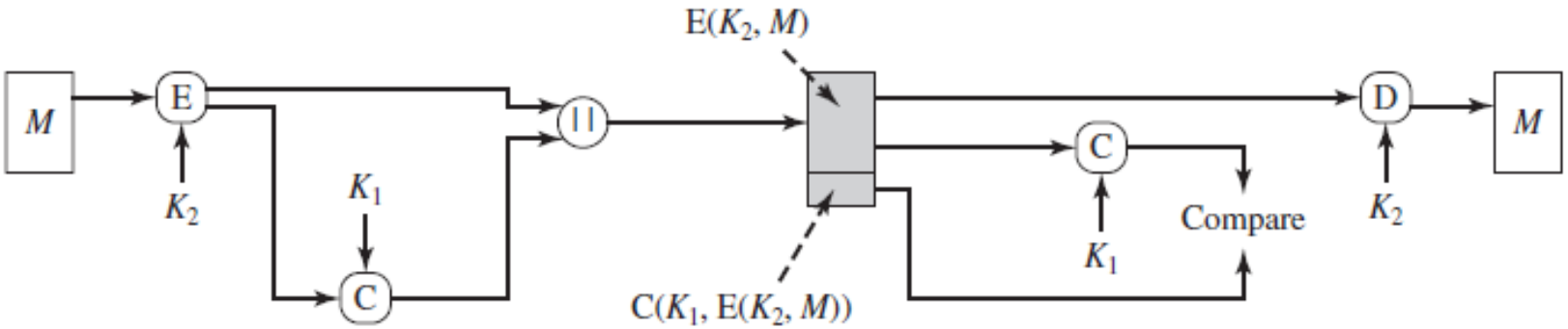
- نگاشت چند به یک است
  - MAC n بیتی: 2^n کد ممکن
  - پیام l بیتی: 2^l پیام ممکن
  - کلید k بیتی: 2^k کلید ممکن
  - هر MAC به طور متوسط توسط 2^{l-n} پیام متفاوت تولید می‌شود و 2^k نگاشت متفاوت از فضای پیام به فضای MAC وجود دارد

کد احراز اصالت پیام و محرمانگی

- دو کلید جداگانه
- معمولا بهتر است که MAC متصل به پیام باشد



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

موارد کاربردی کد احراز اصالت پیام

- چرا همواره از رمزنگاری متقارن (با توجه به کاربرد گسترده آن) برای احراز اصالت استفاده نکنیم؟

1. در کانال پخش با پیام مشترک به همه گیرنده‌ها، یک گیرنده مسئول بررسی اصالت پیام می‌شود و پیام بدون رمز (برای همه گیرنده‌ها) ارسال می‌گردد
2. برای کاهش بار کاری، در برخی کاربردها اصالت برخی پیام‌ها به طور تصادفی بررسی می‌شود
3. احراز اصالت برنامه قابل اجرا (بدون نیاز به رمزگشایی)
4. در برخی کاربردها نیاز به محرمانگی نداریم، مانند SNMPv3 (پیام‌های مدیریتی)
5. جداسازی لایه‌ها: مثلا احراز اصالت در لایه کاربرد و محرمانگی در لایه‌های پایین‌تر
6. در برخی کاربردها می‌خواهیم اصالت پیام همراه با پیام ذخیره شده و در مواقع لازم احراز شود

- توجه: کد احراز اصالت پیام، امضای دیجیتال نیست. چون فرستنده و گیرنده هر دو کلید را می‌دانند

الزامات کد احراز اصالت پیام
مقدمه

- احراز اصالت‌گر با طول ثابت: تابعی از پیام (M) و کلید مخفی (K)
  - برچسب (tag)

$$T = \mathrm{MAC}(K, M)$$

- حمله جستجوی فراگیر به رمزنگاری جهت ایجاد محرمانگی
  - با دانستن متن رمزشده C، امتحان تمام کلیدهای ممکن K_i

$$P_i = \mathrm{D}(K_i, C)$$

- MAC بدون محرمانگی: مهاجم متن اصلی و MAC آن را می‌بیند (T_1, M_1)
  - اگر طول کلید بیشتر از طول برچسب باشد ($k > n$)

$$T_i = \mathrm{MAC}(K_i, M_1) \rightarrow T_i = T_1$$

  - تعداد کلیدها = 2^k > تعداد برچسب‌های متفاوت = 2^n
  - 2^{k-n} کلید برچسب دلخواه را تولید می‌کنند ← تکرار حمله

الزامات کد احراز اصالت پیام
مقدمه

- **Round 1**

Given: $M_1, T_1 = \text{MAC}(K, M_1)$
Compute $T_i = \text{MAC}(K_i, M_1)$ for all 2^k keys
Number of matches $\approx 2^{(k-n)}$

- **Round 2**

Given: $M_2, T_2 = \text{MAC}(K, M_2)$
Compute $T_i = \text{MAC}(K_i, M_2)$ for the $2^{(k-n)}$ keys resulting from Round 1
Number of matches $\approx 2^{(k-2\times n)}$

- α دور لازم است $k = \alpha \times n$

- حمله جستجوی فراگیر به کلید احراز اصالت (کمی) پیچیده‌تر از حمله به کلید رمزگشایی با همان طول است

الزامات کد احراز اصالت پیام
مقدمه

- الگوریتم MAC زیر را در نظر بگیرید:
  - قالب‌های ۶۴ بیتی X_i

$$M = (X_1 \| X_2 \| \ldots \| X_m)$$

- E: الگوریتم DES در سبک ECB
  - کلید ۵۶ بیتی
  - برچسب ۶۴ بیتی

$$\Delta(M) = X_1 \oplus X_2 \oplus \cdots \oplus X_m$$
$$\mathrm{MAC}(K, M) = \mathrm{E}(K, \Delta(M))$$

- مهاجم: $\{M \| \mathrm{MAC}(K, M)\}$
  - حمله جستجوی فراگیر: 2^{56} رمزگذاری
  - جاگذاری X_1 تا X_{m-1} با مقادیر Y_1 تا Y_{m-1} دلخواه و جاگذاری X_m با

$$Y_m = Y_1 \oplus Y_2 \oplus \cdots \oplus Y_{m-1} \oplus \Delta(M)$$

  - پیام $Y_1 \| Y_2 \| \cdots \| Y_m$ اصالت دارد!
  - هر پیام غیرقانونی به طول $64(m\text{-}1)$ بیت را می‌توان درج کرد

الزامات کد احراز اصالت پیام

- فرض: مهاجم MAC را می‌داند ولی کلید را نمی‌داند

1. محاسبه پیام M' برای مهاجم با داشتن $\mathrm{MAC}(K,M)$ و M از نظر محاسباتی غیر ممکن باشد:
$$\mathrm{MAC}(K, M') = \mathrm{MAC}(K, M)$$
2. توزیع $\mathrm{MAC}(K,M)$ یکنواخت باشد: برای پیام‌های تصادفی M و M' :
$$\Pr[\mathrm{MAC}(K, M) = \mathrm{MAC}(K, M')] = 2^{-n}$$
   - مقابله با حمله جستجوی فراگیر بر اساس متن اصلی منتخب
   - توزیع یکنواخت: $2^{(n-1)}$ آزمون برای پیدا کردن پیام متناظر با برچسب داده شده
3. اگر M و M' رابطه مشخصی داشته باشند (یعنی $M'=f(M)$)
$$\Pr[\mathrm{MAC}(K, M) = \mathrm{MAC}(K, M')] = 2^{-n}$$
   - الگوریتم نسبت به برخی بیت‌ها ضعیف نباشد
   - مهاجم با تغییر بیت‌های مشخصی نتواند به پیامی جدید با برچسب قبلی برسد

حمله جستجوی فراگیر

- حمله به فضای کلید
  - حداقل از مرتبه 2^k
- حمله به مقدار MAC (برچسب)
  - یافتن مقدار برچسب بدون دانستن کلید
  - هدف: یافتن برچسب معتبر برای یک پیام و یا یافتن پیامی که متناظر با برچسب باشد (یافتن برخورد)
  - با توجه به خواص MAC: از مرتبه 2^n
  - در تابع چکیده‌ساز (بدون کلید) قابل پیاده‌سازی برون‌خط (offline)
  - در MAC: نیاز به زوج متن اصلی – برچسب منتخب به صورت برخط (online)

$$\min\left(2^k, 2^n\right) \qquad \min(k,n) \geq N \simeq 128\text{bits}$$

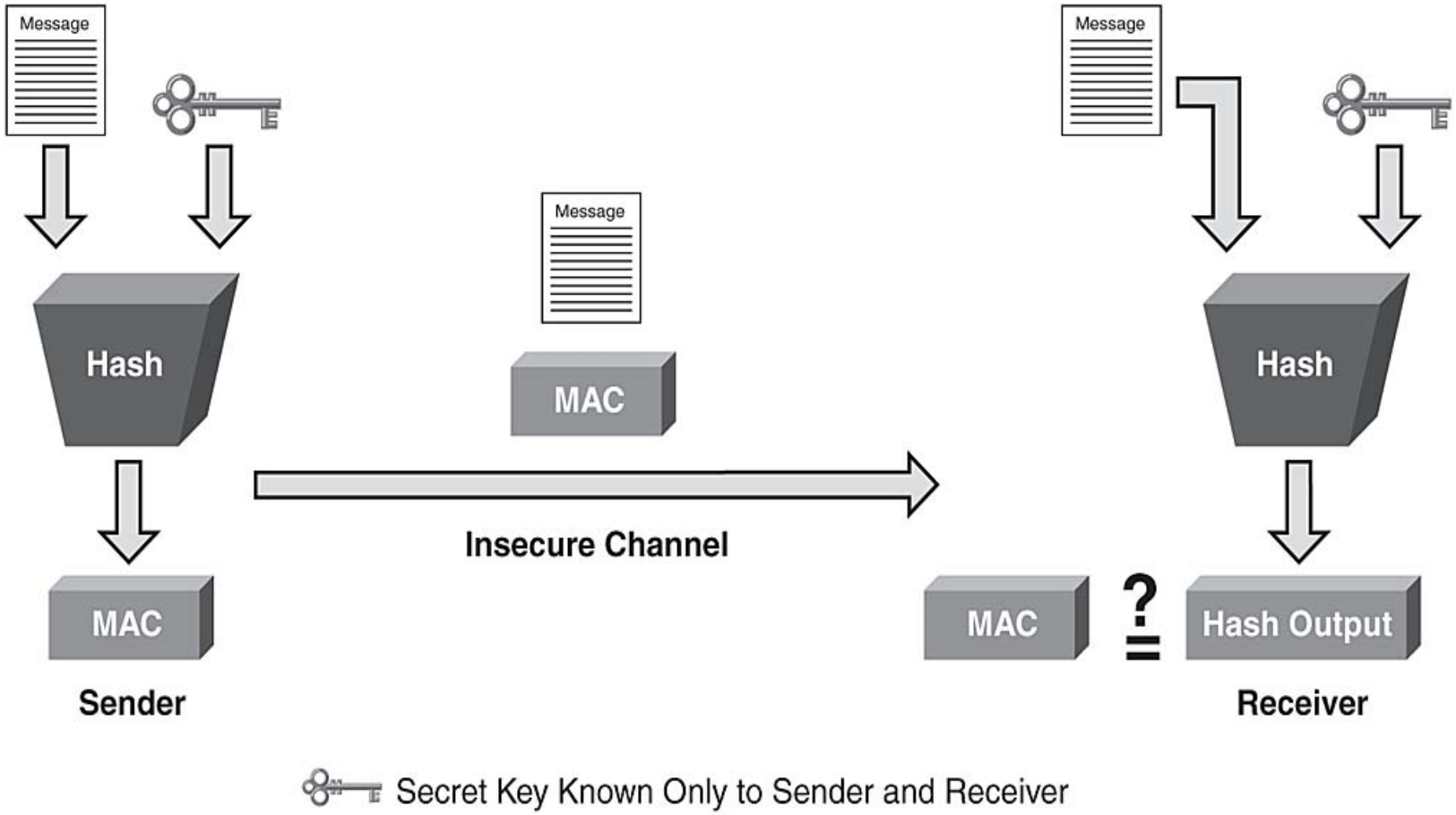
کد احراز اصالت پیام (MAC)

- MAC بر اساس رمز قالبی: روش سنتی معمول
- MAC بر اساس توابع چکیده‌ساز (HMAC)
  - اخیرا مورد توجه زیادی قرار گرفته است
  - سرعت زیاد الگوریتم‌های توابع چکیده‌ساز (مثل MD5 و SHA) نسبت به رمزهای قالبی (مثل DES) و استفاده گسترده از آن‌ها
  - تابع چکیده‌ساز وابسته به کلید نیست و به تنهایی نمی‌توان به عنوان MAC به کار برد
- ترکیب کلید و MAC ← HMAC
  - الزام پیاده‌سازی در امنیت IP
  - استفاده در پروتکل‌های دیگر اینترنت مثل SSL
  - استاندارد NIST ← FIPS 198

HMAC

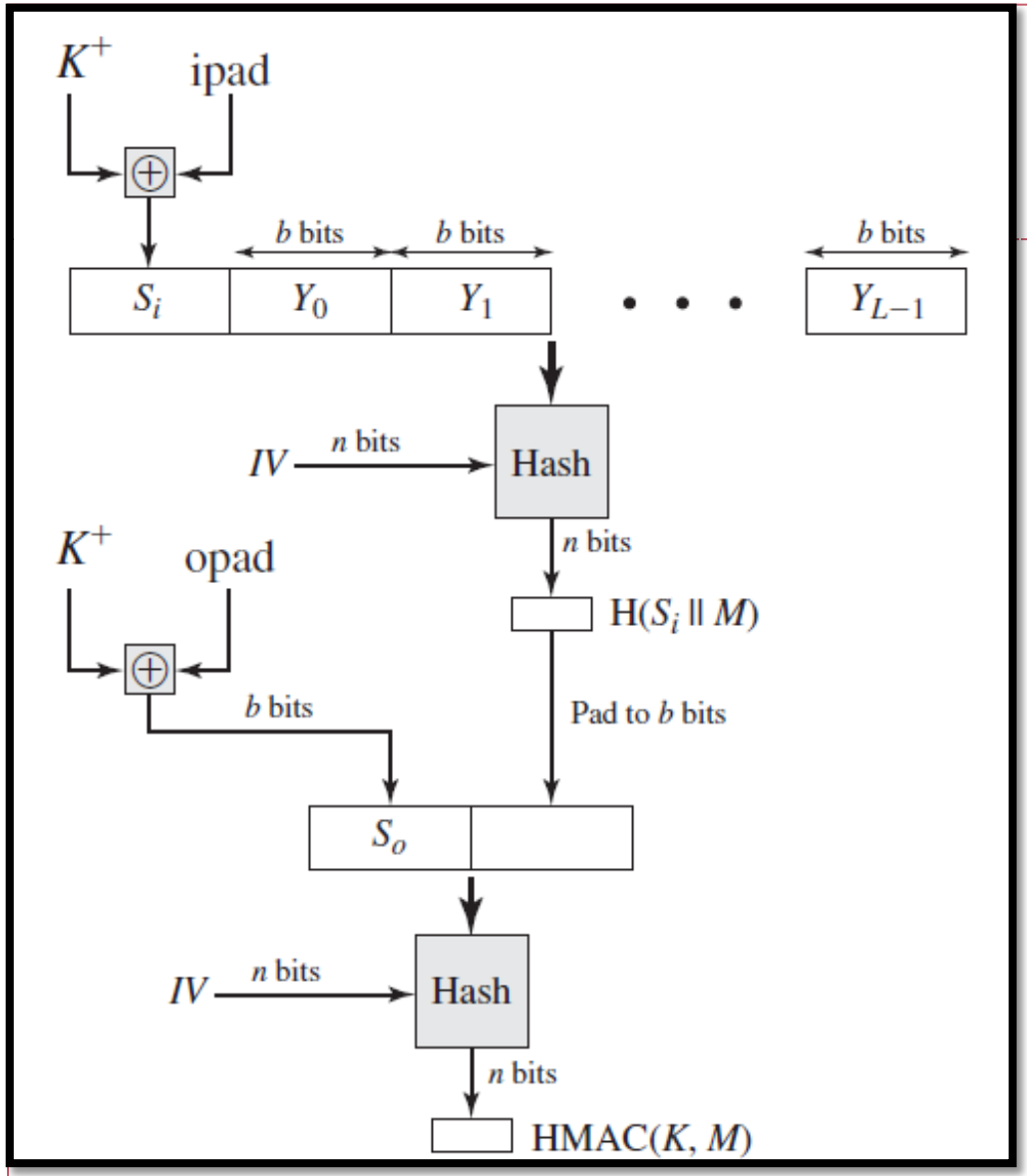
- استفاده از توابع چکیده‌ساز موجود (بدون تغییر) به عنوان هسته اصلی الگوریتم
- جایگزینی ساده تابع چکیده‌ساز در صورت نیاز برای استفاده از تابع چکیده‌ساز بهتر
- حفظ کارآیی تابع چکیده‌ساز
- استفاده ساده از کلید
- تحلیل ساده امنیتی HMAC بر اساس امنیت تابع چکیده‌ساز

HMAC



src: networkworld

ساختار الگوریتم HMAC



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

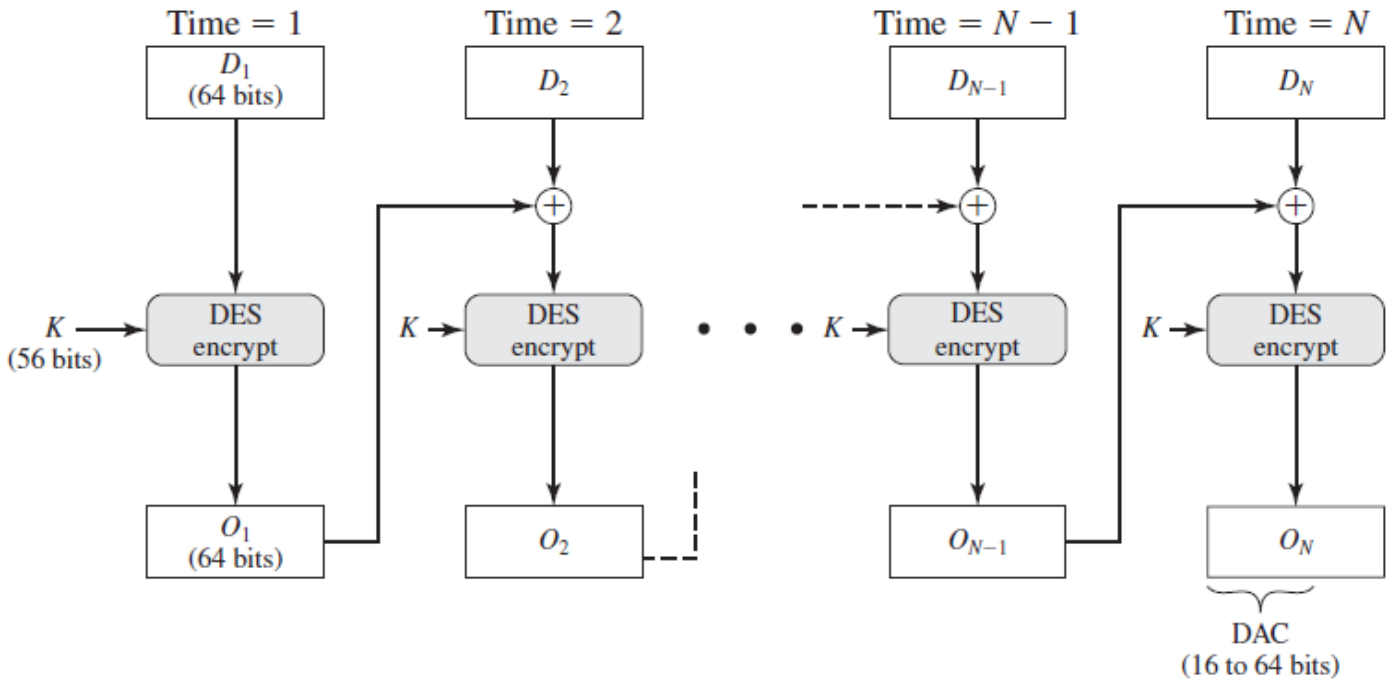
- H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160)
- IV = initial value input to hash function
- M = message input to HMAC
- Y_i = ith block of M
- K^+ = K padded with zeros on the left so that the result is b bits in length
- ipad = 00110110 (36 in hexadecimal) repeated $b/8$ times
- opad = 01011100 (5C in hexadecimal) repeated $b/8$ times

امنیت HMAC

- معادل امنیت تابع چکیده‌ساز استفاده شده
- حمله روز تولد به تابع چکیده‌ساز
  - یافتن دو پیام M و M' که: $H(M) = H(M')$
  - برای چکیده به طول n بیت از مرتبه $2^{n/2}$
  - MD5 ۱۲۸ بیتی؟
- MD5 برای HMAC مناسب است
  - با توجه به استفاده از کلید، یافتن برخورد به طور برون‌خط ممکن نیست و نیاز به 2^{64} قالب (2^{72} بیت) تولید شده توسط یک کلید دارد
  - در یک ارتباط با سرعت 1-Gbps، ۱۵۰۰۰۰ سال طول می‌کشد!
  - اگر سرعت بالا: استفاده از MD5 به جای SHA

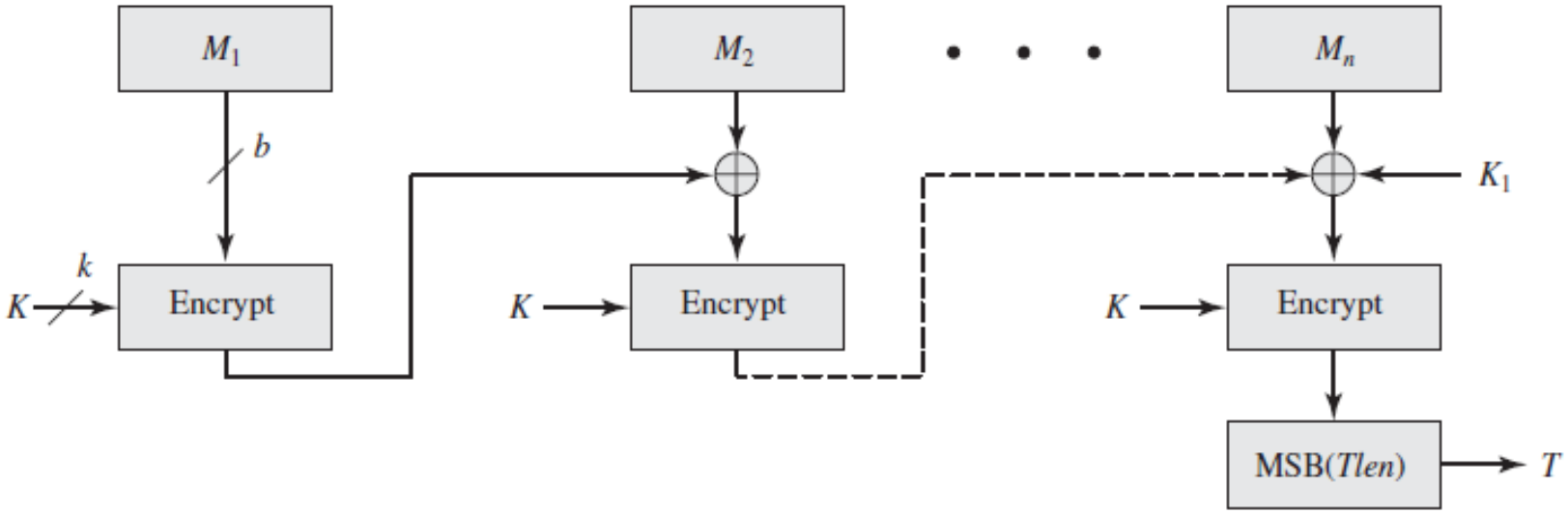
MAC بر اساس رمز قالبی

- DAA (Data Authentication Algorithm): بر اساس DES در سبک CBC
  - خروجی = data authentication code (DAC)
- مدت‌ها پرکاربردترین الگوریتم MAC بوده
  - استاندارد NIST و ANSI (X9.17)



MAC بر اساس رمز قالبی

- بهبود الگوریتم به دلیل مشکلات امنیتی DAA
  - استفاده از AES و 3-DES
  - Cipher-based Message Authentication Code (CMAC)



رمزگذاری توام با احراز اصالت
Authenticated encryption (AE)

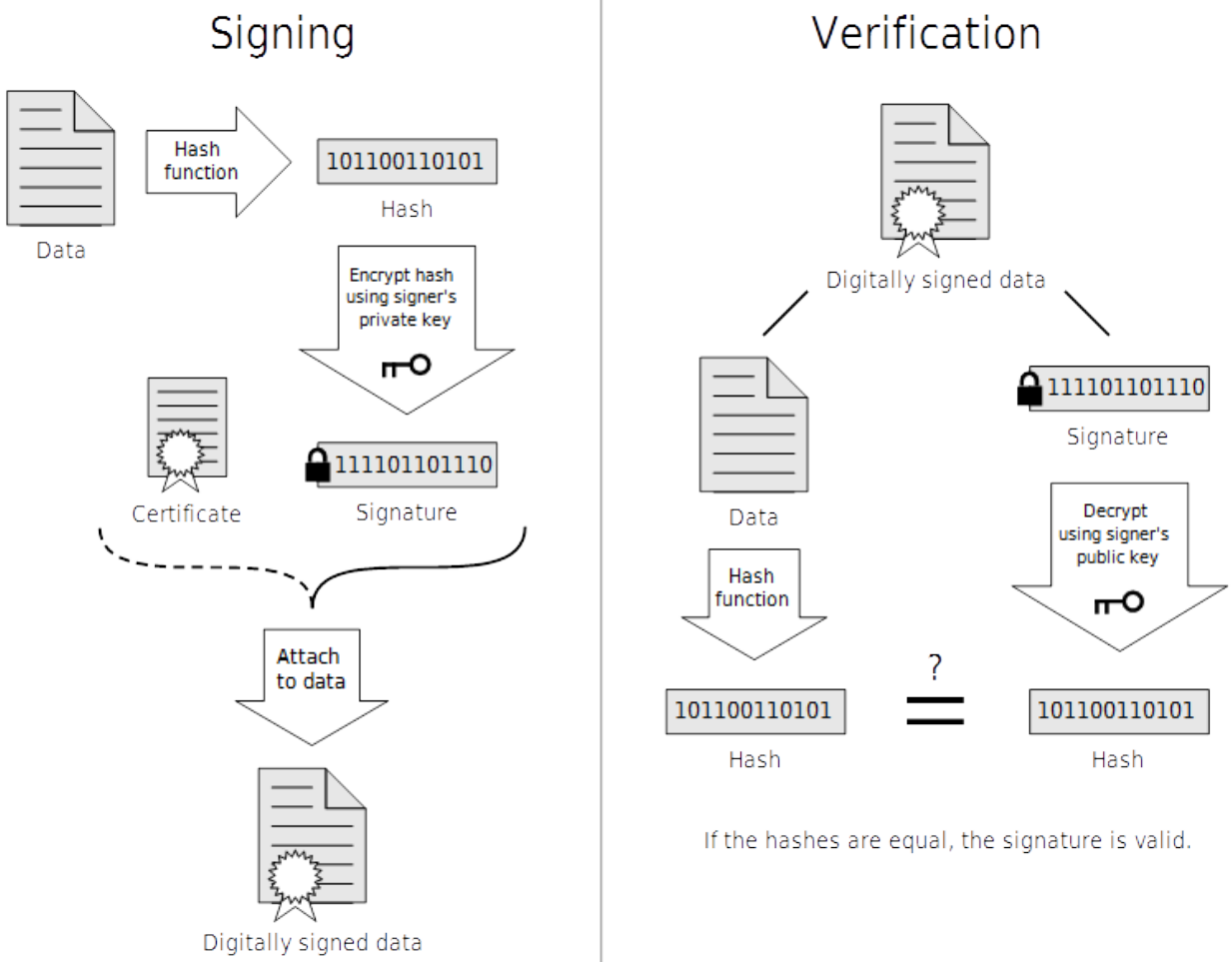
- محرمانگی و احراز اصالت توام (اخیرا)

- Hashing followed by encryption (H → E)
  - Wired Equivalent Privacy (WEP) protocol in WiFi
- Authentication followed by encryption (A → E)
  - SSL/TLS protocols
- Encryption followed by authentication (E → A)
  - IPSec protocol
- Independently encrypt and authenticate (E + A)
  - SSH protocol
- Counter with cipher block chaining-message authentication code (CCM)
  - IEEE 802.11 WiFi - NIST SP 800-38C
  - Encrypt-and-MAC (AES+CMAC)
- Galois/Counter Mode (GCM)

امضای دیجیتال
Digital Signature

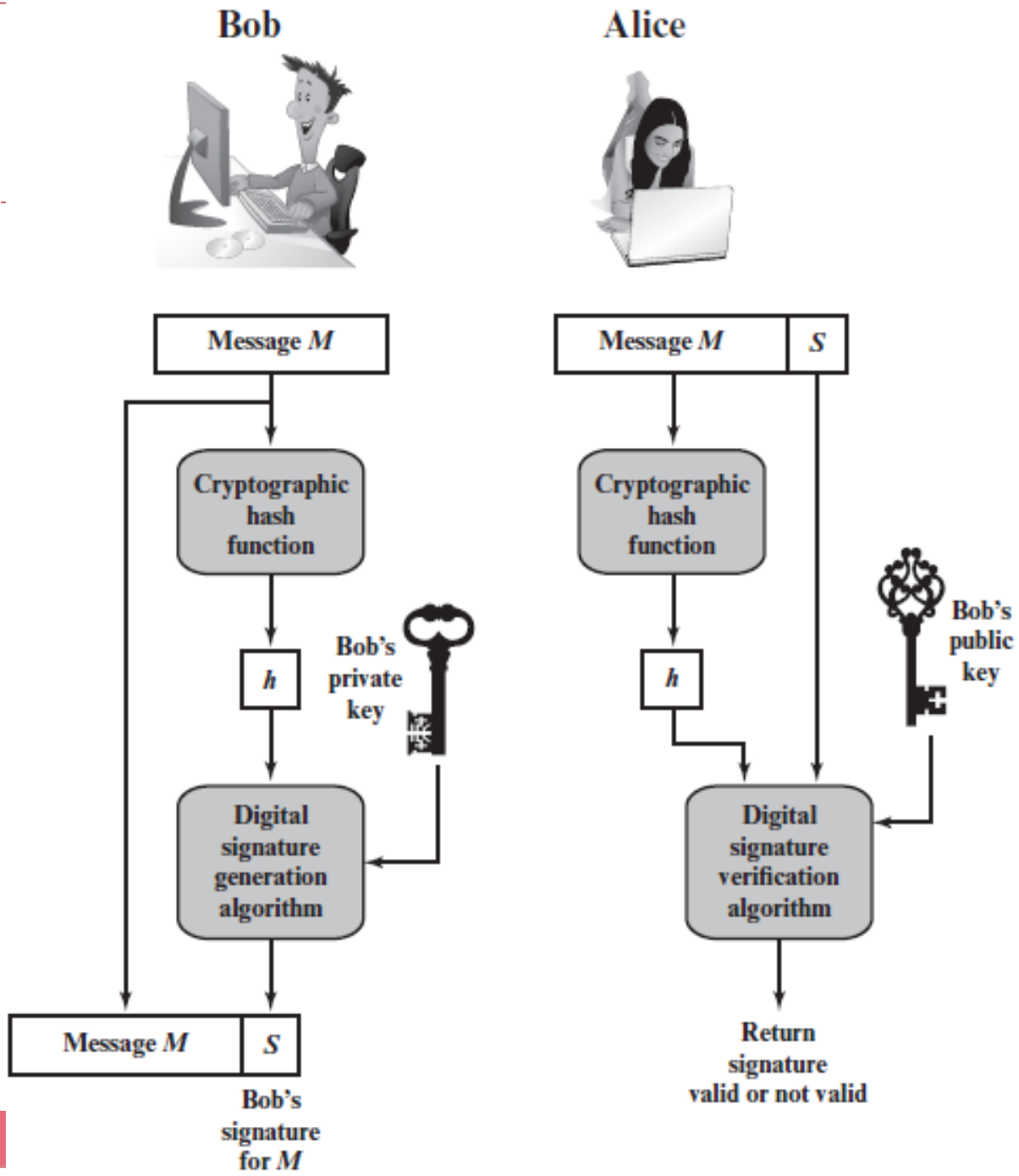
- احراز اصالت پیام از تغییر پیام توسط شخص سوم جلوگیری می‌کند
- چه اختلاف‌هایی ممکن است رخ دهد؟
- جعل توسط گیرنده
  - گیرنده پیام دلخواه خود و کد احراز اصالت آن (با استفاده از کلید مخفی) را تولید کرده و ادعا کند که فرستنده آن را ارسال کرده است
  - مثال: در ارسال‌های مالی، گیرنده مبلغ را افزایش دهد
- انکار توسط فرستنده
  - با توجه به امکان جعل توسط گیرنده، فرستنده پیام ارسالی را انکارکند
  - مثال: دستور خرید سهام در بورس توسط پست الکترونیکی ارسال شود. پس از کاهش ارزش سهام، فرستنده ارسال آن را انکار کند
- راه حل: امضای دیجیتال (بر اساس رمزنگاری کلید همگانی)
  - در شرایطی که اعتماد کامل بین فرستنده و گیرنده وجود ندارد

امضای دیجیتال (Digital Signature)



- فرستنده پیام را امضا می‌کند
- ورودی الگوریتم امضا: پیام و کلید خصوصی فرستنده
- هر گیرنده‌ای می‌تواند امضا را تایید کند
- ورودی الگوریتم تایید: پیام، امضا و کلید همگانی فرستنده

src: wikipedia

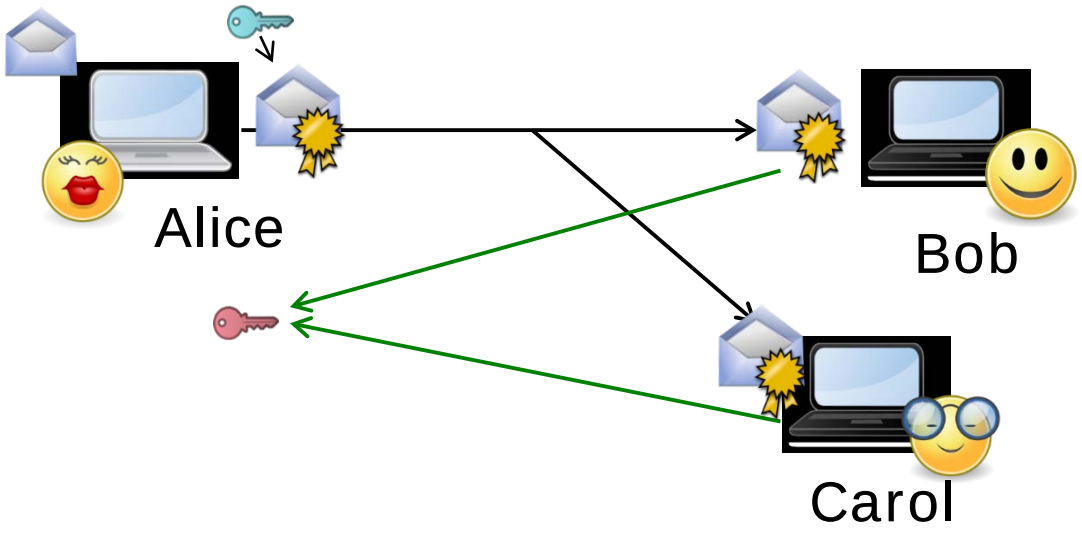

Bob
Alice
Message M
Cryptographic hash function
h
Bob's private key
Digital signature generation algorithm
Message M
S
Bob's signature for M
Message M
S
Cryptographic hash function
h
Bob's public key
Digital signature verification algorithm
Return signature valid or not valid

ویژگی‌های امضای دیجیتال

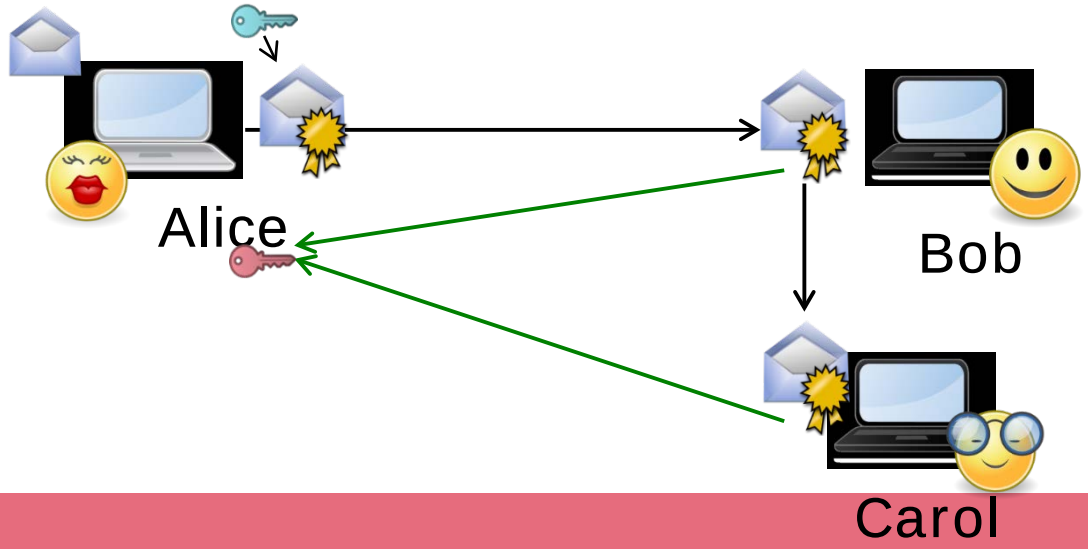
1. تایید شخص امضا کننده، تاریخ و زمان امضا
2. احراز اصالت محتوای پیام در زمان امضا
3. امکان تایید توسط شخص سوم در موارد بروز اختلاف

- امضای دیجیتال در برگیرنده احراز اصالت است

ویژگی‌های امضای دیجیتال



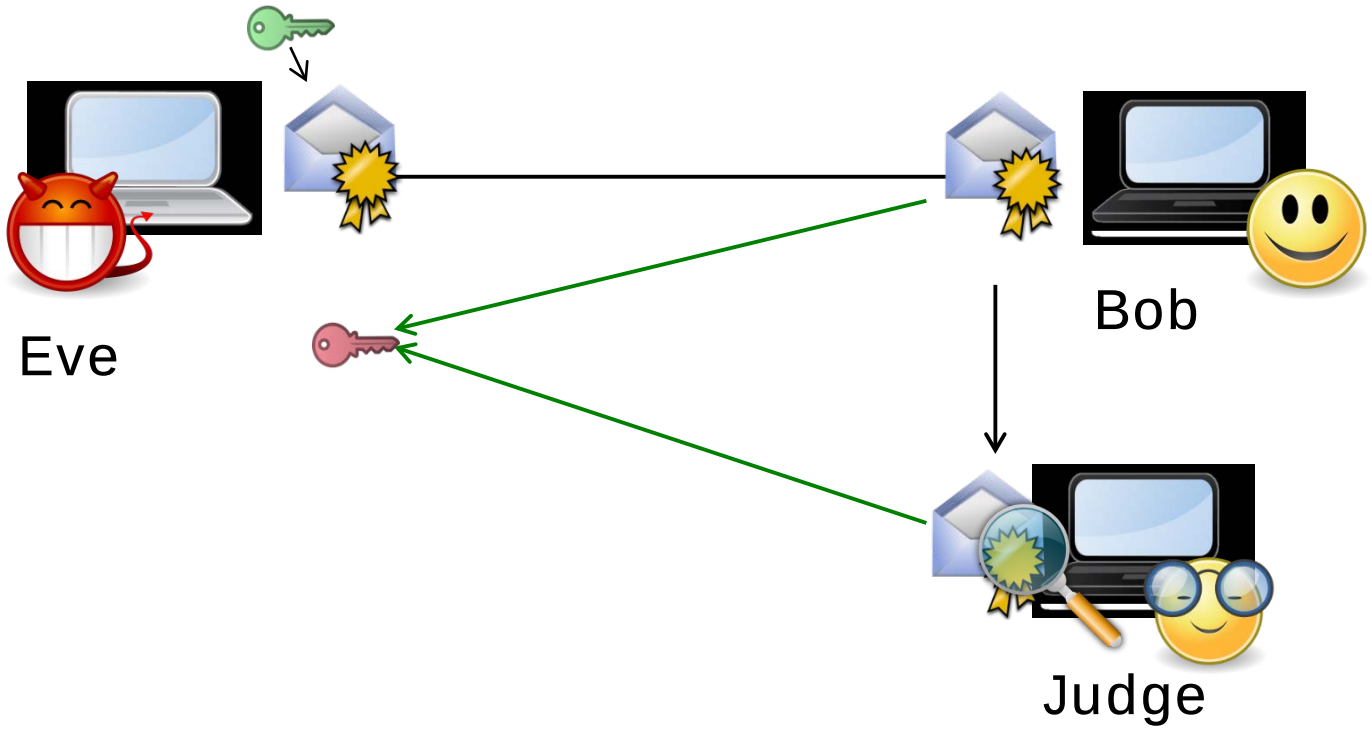
- وارسی پذیری همگانی
  - MAC بدون داشتن کلید وارسی پذیر نیست



- انتقال پذیر
  - وارسی توسط شخص سوم

ویژگی‌های امضای دیجیتال

- انکارناپذیری



حملات امضای دیجیتال

حملات مختلف بر اساس **اطلاعات مهاجم**

- حمله کلید-مبنا (Key-only attack): کلید همگانی امضاکننده
- حمله پیام معلوم (Known message attack): برخی پیام‌ها و امضاهای متناظر
- حمله پیام منتخب (Chosen message attack): پیام‌های منتخب و امضاهای متناظر

اهداف مهاجم

- شکست کامل (Total break): بدست آوردن کلید خصوصی امضاکننده
- جعل فراگیر (Universal forgery): پیداکردن الگوریتمی برای تولید امضای معادل برای پیام دلخواه
- جعل انتخابی (Selective forgery)
- جعل وجودی (Existential forgery): تولید امضای معادل برای حداقل یک پیام

الزامات امضای دیجیتال

- امضا باید دنباله‌ای از بیت‌های باشد که به پیام امضا شده بستگی دارد
- برای تولید امضا باید از اطلاعات منحصر به امضا کننده استفاده شود
  - رمزنگاری کلید همگانی
- تولید امضا ساده باشد
- تشخیص و تایید امضا ساده باشد
- جعل امضا از نظر محاسباتی غیر ممکن باشد
  - تولید پیام جدید برای امضای موجود
  - تولید امضای تقلبی برای پیام دلخواه
- ذخیره امضا در حافظه، جهت بررسی‌های آتی امکان‌پذیر باشد

امضای دیجیتال مستقیم
Direct Digital Signature

- تنها فرستنده و گیرنده دخیل هستند
  - گیرنده، کلید همگانی فرستنده را می‌داند
- محرمانگی: رمزگذاری پیام و امضا با استفاده از یک کلید مخفی
  - امضای دیجیتال متن اصلی (نه رمز شده)
- امنیت این سیستم، وابسته به امنیت کلید خصوصی فرستنده است
  - فرستنده در صورت تمایل به انکار، می‌تواند ادعای سرقت آن را نماید
  - یک راه حل: استفاده از مهر زمانی (timestamp) و درخواست اعلام بی‌درنگ سرقت کلید
  - ممکن است، دشمن مهر زمان‌های قبل از سرقت را روی پیام بزند!
- راه حل: استفاده از گواهی امضا و مرجع مجازشناس امضا
  - مبحث مدیریت کلید

امضای دیجیتال Elgamal
تولید

- عدد اول q و ریشه اولی α
- تولید کلیدها در کاربر A

1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

- چکیده $m = \mathrm{H}(M)$ $\quad 0 \le m \le q - 1$

1. Choose a random integer K such that $1 \le K \le q - 1$ and $\gcd(K, q - 1) = 1$. That is, K is relatively prime to $q - 1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q - 1)$. That is, compute the inverse of K modulo $q - 1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1)$.
5. The signature consists of the pair (S_1, S_2).

امضای دیجیتال Elgamal
تایید

- هر کاربر B

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

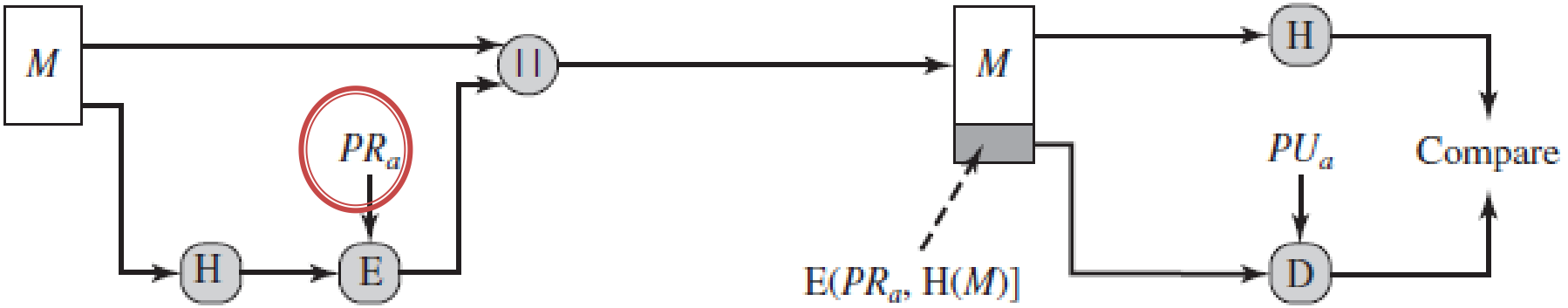
- تایید امضا: $V_1 = V_2$

$$\alpha^m \bmod q = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$$
$$\alpha^m \bmod q = \alpha^{X_A S_1}\alpha^{KS_2} \bmod q$$
$$\alpha^{m-X_A S_1} \bmod q = \alpha^{KS_2} \bmod q$$
$$m - X_A S_1 \equiv KS_2 \bmod (q-1)$$
$$m - X_A S_1 \equiv KK^{-1}(m - X_A S_1) \bmod (q-1)$$

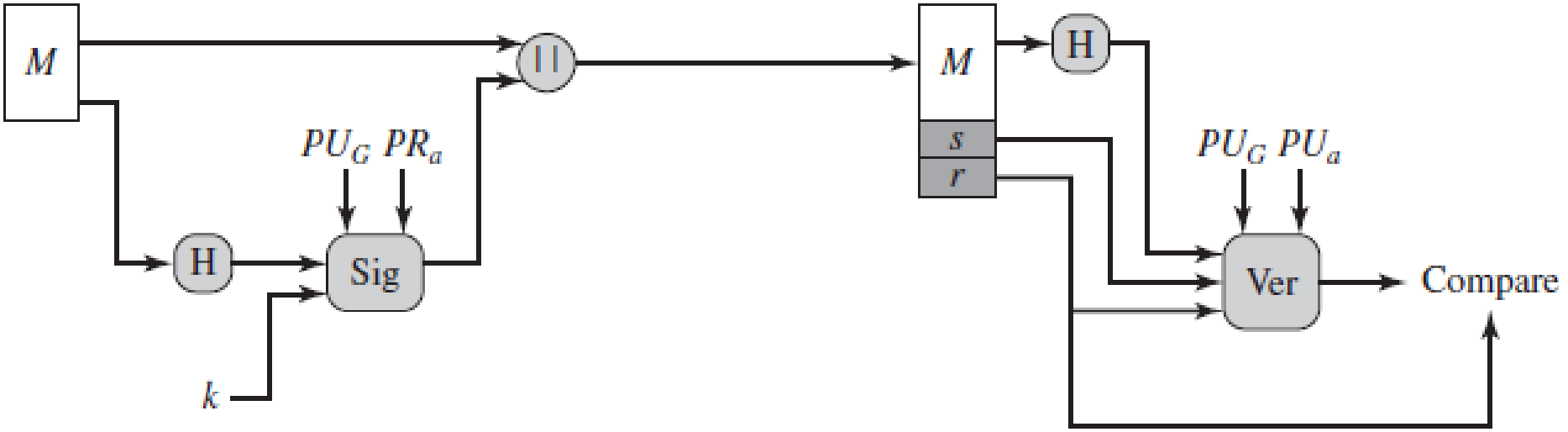
استانداردهای امضای دیجیتال

- Digital Signature Standard (DSS)
  - استاندارد NIST (FIPS 186)
  - بکارگیری Secure Hash Algorithm (SHA)
  - الگوریتم تولید امضا: Digital Signature Algorithm (DSA)
    - بر پایه الگوریتم‌های ElGamal 1985 و Schnorr 1991
  - ۱۹۹۱، ۱۹۹۶، ۲۰۰۰، ۲۰۰۹، ۲۰۱۳ ← FIPS 186-4
  - در نسخه نهایی، بکارگیری RSA یا رمز elliptic curve نیز ممکن است
- استاندارد امضای دیجیتال بر اساس RSA
  - ISO 9776
  - ANSI X9.31
  - CCITT X.509

استانداردهای امضای دیجیتال



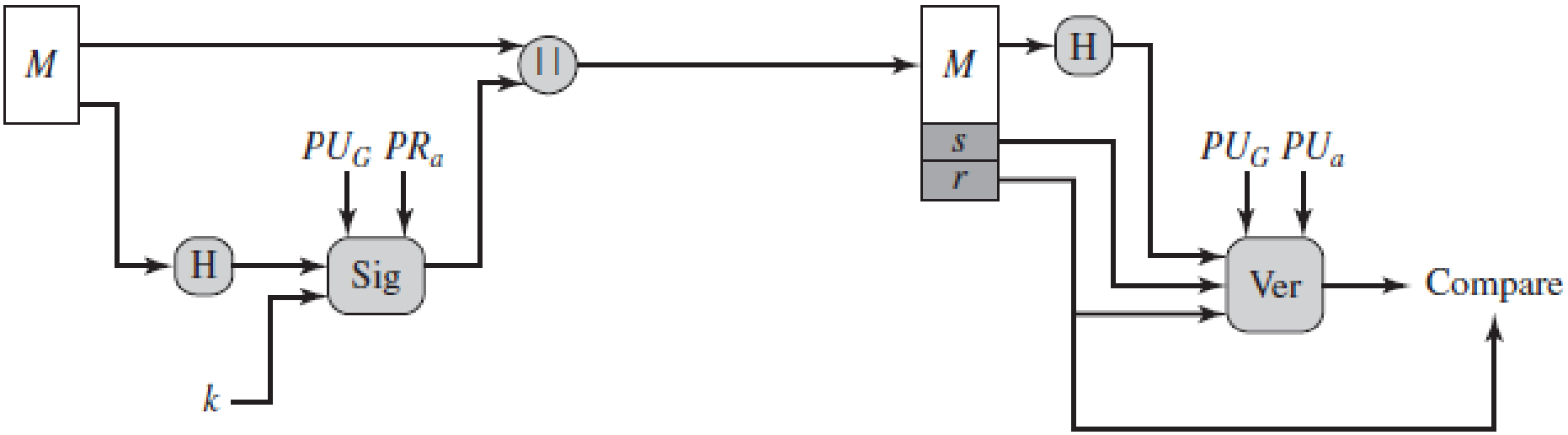
(a) RSA approach



(b) DSS approach

DSS

- تابع چکیده‌ساز
- تابع امضا: ورودی‌ها
  - چکیده پیام، کلید خصوصی فرستنده، مقدار تصادفی k (تولید شده در هر بار امضا)
  - کلید همگانی سراسری global public key (PU_G)
- خروجی تابع امضا: دو امضا (s و r)
- برخلاف RSA، در رمزنگاری و توزیع کلید کاربرد ندارد



الگوریتم تولید امضا
Digital Signature Algorithm (DSA)

Global Public-Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$ and L a multiple of 64; i.e., bit length L between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p - 1)$, where $2^{N-1} < q < 2^N$ i.e., bit length of N bits

g $= h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p - 1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's Per-Message Secret Number

k random or pseudorandom integer with $0 < k < q$

- بر اساس سختی محاسبه لگاریتم گسسته
- بر پایه الگوریتم‌های ElGamal 1985 و Schnorr 1991
- ۳ پارامتر کلی: کلید همگانی سراسری
- کلید خصوصی و همگانی امضا کننده (رابطه یک‌طرفه)

الگوریتم تولید امضا
Digital Signature Algorithm (DSA)

Signing

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1}(H(M) + xr)] \bmod q$

Signature $= (r, s)$

Verifying

$w = (s')^{-1} \bmod q$

$u_1 = [H(M')w] \bmod q$

$u_2 = (r')w \bmod q$

$v = [(g^{u1}y^{u2}) \bmod p] \bmod q$

TEST: $v = r'$

M = message to be signed

H(M) = hash of M using SHA-1

M', r', s' = received versions of M, r, s

- دو امضا (s و r)
  - تایید بر روی r صورت می‌گیرد که تنها تابعی از مقدار تصادفی و کلید همگانی سراسری است (مستقل از پیام)
  - s تابعی از کلید خصوصی فرستنده است و گیرنده را قادر می‌سازد که بدون دانستن مقدار تصادفی، r را تایید کند

از لحاظ محاسباتی کارآ است و قسمت‌های پیچیده آن مستقل از پیام هستند