

کدهای تصدیق صحت پیام و توابع درهم ساز

MAC codes and Hash functions

اصغراصل اصغریان - دانشگاه ارومیه

فهرست مطالب

🔒 مفاهیم اولیه

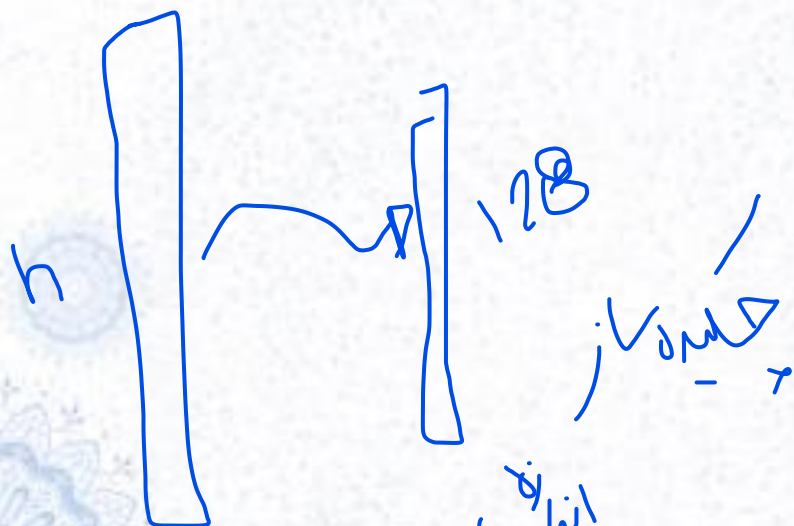
🔒 رمزگذاری پیام و کدهای تشخیص خطا

🔒 کدهای تصدیق صحت پیام

🔒 اصول توابع درهم ساز

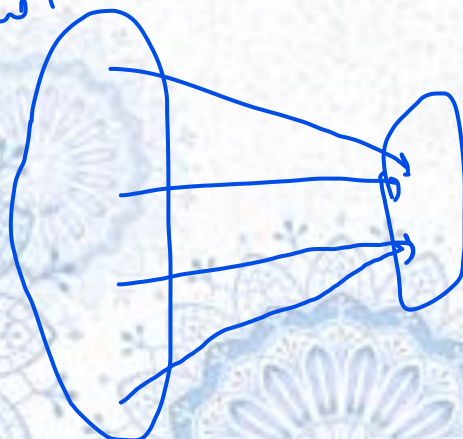
🔒 توابع درهم ساز مهم

🔒 HMAC



حاشیه ساز

انرژی دافنه خشی بزرگ



انرژی دافنه خشی بزرگ

تصدیق صحت پیام چیست؟

🔒 اطمینان از صحت محتوای پیام (Data Integrity)

🔒 یعنی پیام دریافتی دستکاری نشده باشد

▪ بدون تغییر

▪ بدون درج

▪ بدون حذف

🔒 پیام از جانب فرستنده ادعا شده ارسال شده است (Origin Integrity)

تصدیق صحت پیام

🔒 در برخی از کاربردها:

- 🔒 حفظ محرمانگی پیام اهمیت بالایی ندارد
- 🔒 ولی صحت آنها اهمیت بالایی دارد (قابل اعتماد بودن پیام)
- 🔒 مثال: تراکنش‌های بانکی

🔒 نیاز به دو عملکرد داریم:

- 🔒 یک تابع برای تولید عامل تصدیق کننده
- 🔒 یک تابع برای واریسی (یا چک کردن) عامل تصدیق کننده

فهرست مطالب

❏ مفاهیم اولیه

❏ رمزگذاری پیام و کدهای تشخیص خطا

❏ کدهای تصدیق صحت پیام

❏ اصول توابع درهم ساز

❏ توابع درهم ساز مهم

HMAC ❏

رمز گذاری پیام برای تصدیق صحت پیام

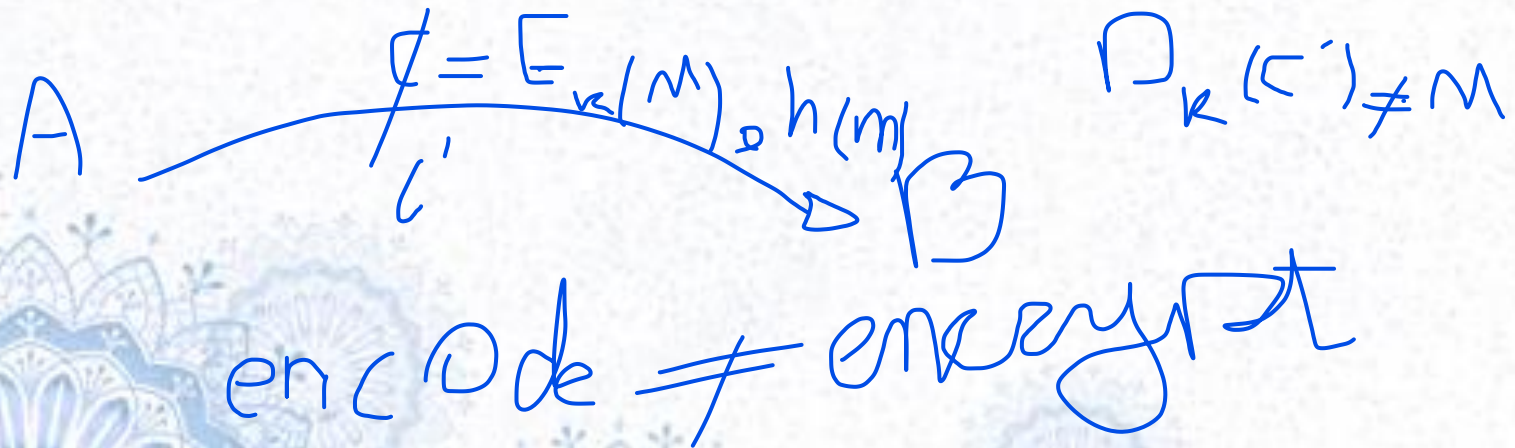
❗ فرستنده پیام را رمز می کند

❗ اگر متن رمز شده دستکاری شود

❗ با رمز گشایی به متن آشکار نامفهوم (درهم و برهم) می رسیم

❗ گیرنده، بعد از رمز گشایی چک می کند ک آیا پیام مفهوم است یا نه؟

❗ می توان از الگوریتم های رمز متقارن و یا نامتقارن برای این منظور استفاده کرد



مشکلات استفاده از رمزنگاری

❏ کارایی پایین

❏ بررسی مفهوم بودن محتوا همواره آسان نیست

❏ برای مثال باید پیام یک قالب استاندارد داشته باشد

❏ نیاز به افزودن

❏ دشواری خودکارسازی فرآیند های تولید و واریسی

❏ راه حل اولیه: استفاده از **کدهای تشخیص خطا**

❏ مثال ساده: استفاده از بیت Parity:

▪ یک بیت به انتهای پیام اضافه نماییم، به گونه ای که تعداد بیت های یک، زوج شود

کدهای تشخیص خطا

۳۲ بیت ۳۲ بیت همه در جمع کنه

له غزری
32 بیت

❏ فرض کنید F یک تابع برای تولید کد تشخیص خطا باشد

❏ یک مثال از تابع F کد CRC است:

CRC: Cyclic Redundancy Check ❏

❏ مثال:

$\text{CRC-32("Hello")} = 0xF7D18982$

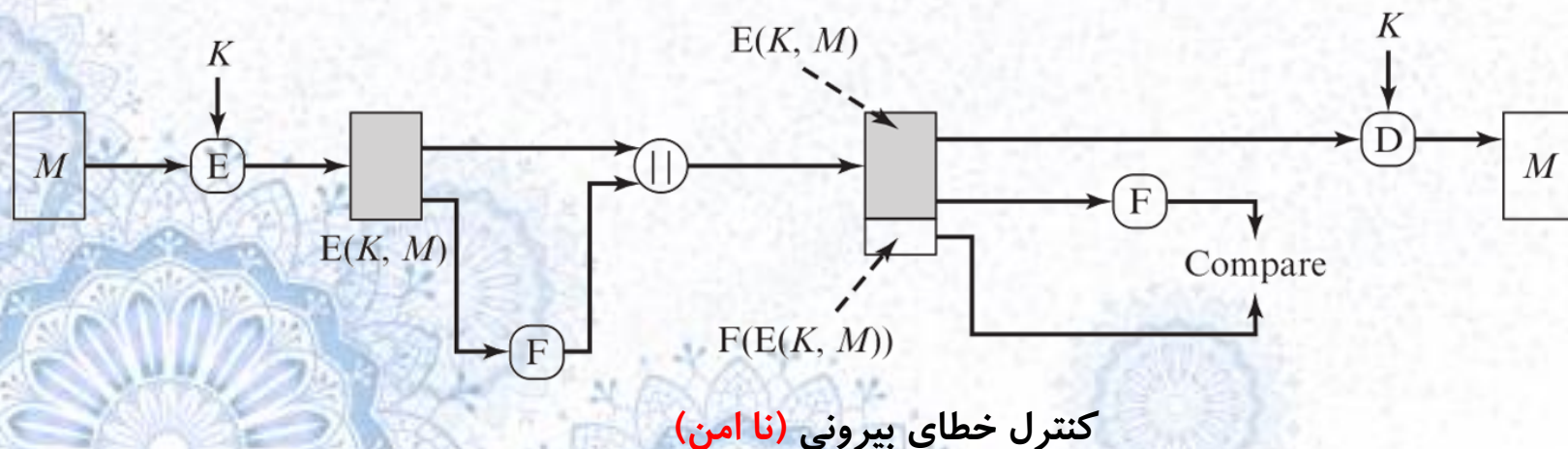
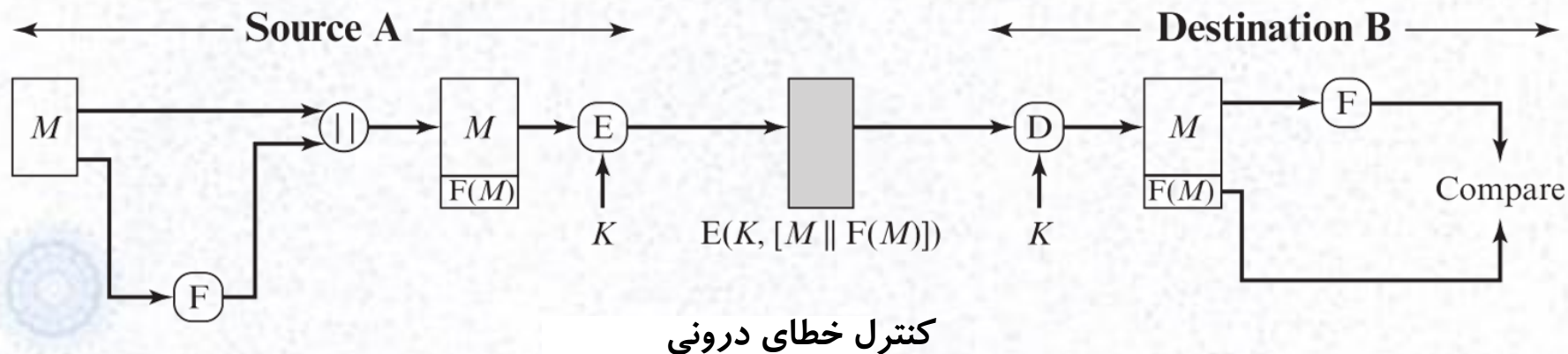
❏ کد تشخیص خطا به عنوان **برچسب پیام** همراه پیام رمز شده ارسال می شود

❏ گیرنده، بعد از رمز گشایی چک می کند

❏ که آیا «کد تشخیص خطای» محاسبه شده با استفاده از F با برچسب پیام

مطابقت دارد یا نه؟

انواع روشهای اعمال کدهای تشخیص خطا



ناآمن 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 برای کاربردهای

امنیتی استفاده می‌کردند، صورت پذیرفته است:

❏ مثال: پروتال WEP

جمع بندی

- ❏ هدف رمزگذاری، **محرمانگی** است و نه صحت
- ❏ کد تشخیص خطا از هیچ کلیدی استفاده نمی کند
- ❏ کد تشخیص خطا نمی تواند در حالت کلی از دستکاری بسته ها جلوگیری کند
- ❏ راه حل: کدهای **تصدیق صحت** پیام

فهرست مطالب

- ❏ مفاهیم اولیه
- ❏ رمزگذاری پیام و کدهای تشخیص خطا
- ❏ کدهای تصدیق صحت پیام
- ❏ اصول توابع درهم ساز
- ❏ توابع درهم ساز مهم
- ❏ HMAC

کدهای تصدیق صحت پیام

MAC: Message Authentication Code 🔒

نام دیگر: Cryptographic Checksum 🔒

هدف: تولید یک برچسب (Tag) با طول ثابت: 🔒

وابسته به پیام 🔒

لزوماً برگشت پذیر نیست (بر خلاف توابع رمزنگاری) 🔒

نیازمند اشتراک یک کلید مخفی بین طرفین 🔒

کدهای تصدیق صحت پیام (۲)

- گیرنده و فرستنده‌ها یک **کلید مشترک** دارند
- فرستنده با استفاده از کلید یک **برچسب** برای هر پیام تولید می‌کند
- فرستنده برچسب تولید شده را به پیام **الحاق** می‌کند (همراه پیام ارسال می‌کند)
- گیرنده پیام، برچسب، و کلید را به الگوریتم **تصدیق** (Verification) می‌دهد
- الگوریتم تصدیق، برچسب را دوباره با کلید مشترک محاسبه می‌کند و با برچسب ارسالی مقایسه می‌کند
- در صورتی که خروجی الگوریتم TRUE باشد، از صحت پیام و هویت فرستنده اطمینان حاصل می‌کند

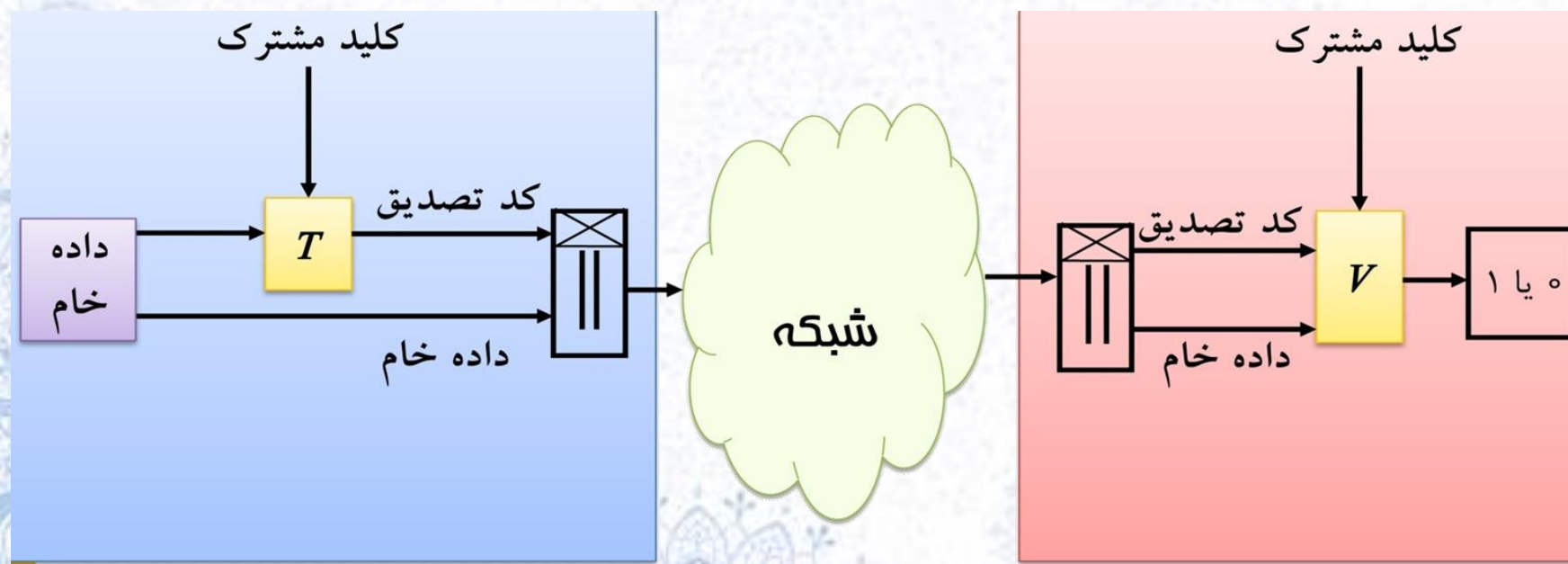
نحوه عملکرد کدهای تصدیق صحت پیام

T : Tag

V : Verification

حسن

علی



سه روش برای ترکیب MAC با رمزنگاری

AtE (Authenticate then Encrypt) 🔒

$$E_{K2}(M \parallel T_{K1}(M)) \quad \times$$

EtA (Encrypt then Authenticate) 🔒

$$E_{K2}(M) \parallel T_{K1}(E_{K2}(M)) \quad \checkmark$$

E&A (Encrypt & Authenticate) 🔒

$$E_{K2}(M) \parallel T_{K1}(M) \quad \times$$

🔒 **K1** کلید تصدیق صحت و **K2** کلید رمزنگاری (محرمانگی)

توضیح در مورد سه روش

🔒 در حالت کلی، فقط روش EtA امن است

🔒 در حالت‌های خاص ممکن است روش‌های دیگر هم امن باشند

🔒 IPsec از این روش استفاده می‌کند

🔒 SSL از حالت خاصی از AtE استفاده می‌کند که امن است

🔒 SSH از حالت خاصی از E&A استفاده می‌کند که امن است

H. Krawczyk. The Order of Encryption and Authentication 🔒 for Protecting Communications (Or How Secure Is SSL), CRYPTO 2001.

ساختن MAC امن با استفاده از توابع رمز گذاری

🔒 تابع MAC:

🔒 با استفاده از پیام و کلید (ورودیها) یک کد درست می کند

🔒 تفاوت با تابع رمزنگاری:

▪ لزوماً برگشت پذیر نیست (به همین علت امن تر است)

🔒 با استفاده از توابع رمز گذاری امن و برخی از سبک های رمزنگاری می توان توابع MAC امن ساخت

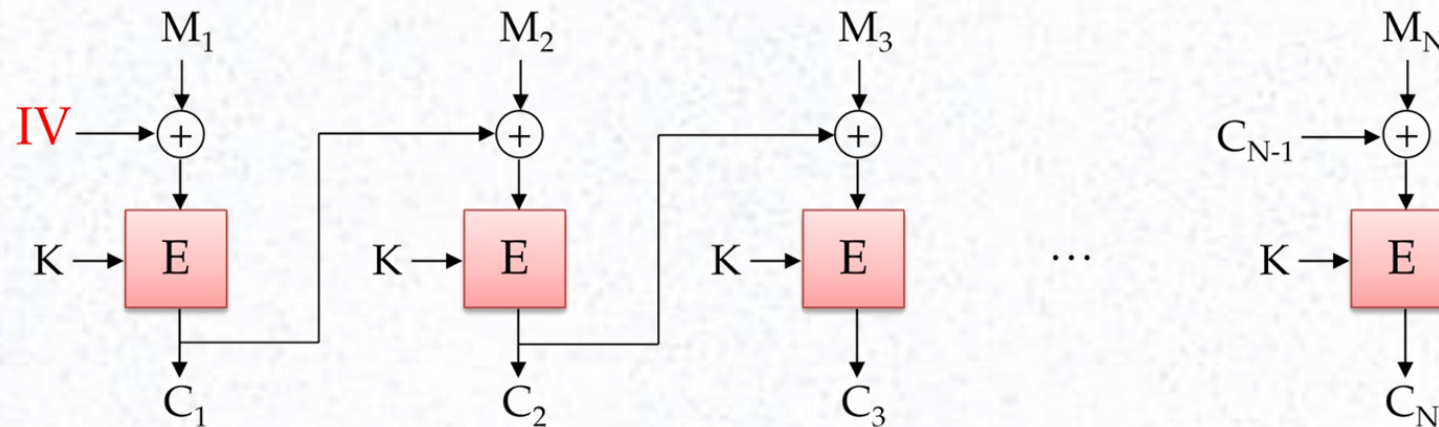
🔒 مثال: سبک های CBC و CFB

▪ در ساختن MAC از این سبک ها باید دقت زیادی کرد

▪ جزئیات بسیار مهم اند

🔒 در ادامه تلاش می کنیم تا با سبک CBC، MAC بسازیم!

CBC-MAC – تلاش ۱



🔒 پیام: $M=(M_1, \dots, M_N)$

🔒 برچسب: $T=(IV, C_N)$

🔒 پیام به همراه برچسب فرستاده می شود

🔒 برای تصدیق، برچسب از نو محاسبه و با برچسب دریافتی مقایسه می شود

حمله به تلاش ۱

❗ مهاجم می تواند با انتخاب IV ، قطعه اول پیغام را به دلخواه تغییر دهد

❗ با داشتن پیام $M=(M_1, M_2, \dots, M_N)$ و برچسب $T=(IV, C_N)$

❗ می توان پیام و برچسب جدیدی را بدون داشتن کلید جعل کرد:

▪ مهاجم قطعه اول پیام را به دلخواه عوض می کند

$$M'=(M_1', M_2, \dots, M_N)$$

▪ و می تواند IV' را به روش زیر حساب کند

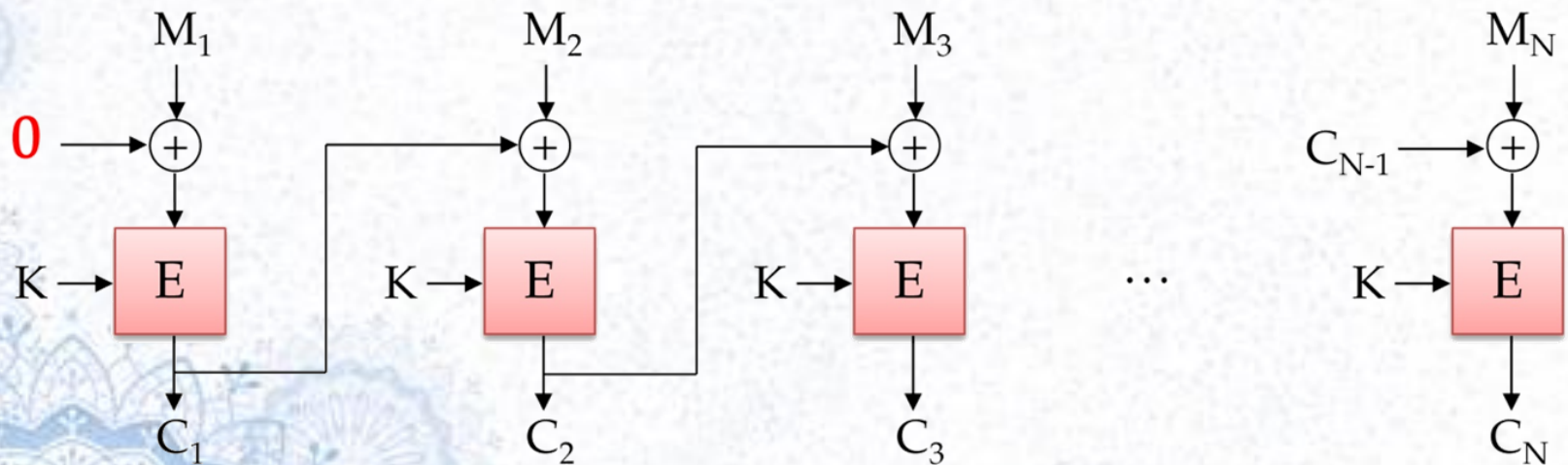
$$IV' \oplus M_1' = IV \oplus M_1 \quad \Rightarrow \quad IV' = IV \oplus M_1 \oplus M_1'$$

▪ برچسب حاصل:

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

CBC-MAC – تلاش ۲

🔒 راهکار حمله به تلاش ۱: استفاده از یک مقدار ثابت برای IV ، مثلاً بردار صفر
🔒 برچسب: مساوی C_N است



حمله به تلاش ۲ – افزایش طول (Length Extension)

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

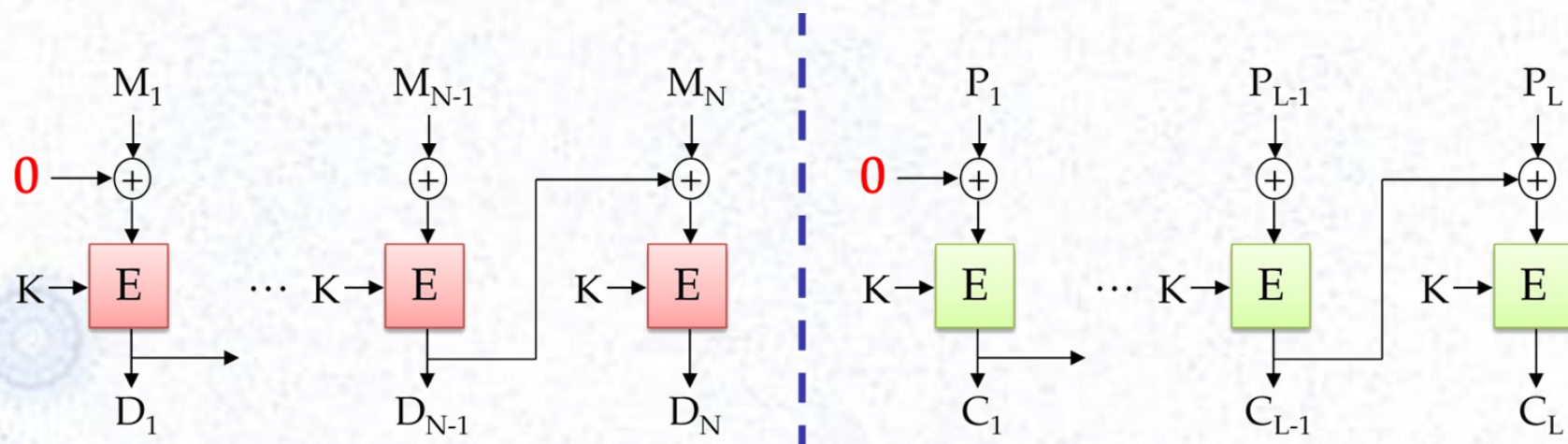
$$M'=M_1, M_2$$

$$T'=T=C_1$$

$$M_2=M_1 \oplus C_1$$

🔒 به همین ترتیب می توان جعل را ادامه داد و به پیام هایی با طول بیشتر رسید

حمله به تلاش ۲ – برچسب جدید از دو برچسب موجود



🔒 دو پیام و برچسب روی هریک را داریم:

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

🔒 پیام $M=(M_1, \dots, M_N)$ با برچسب D_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 حمل افزایش طول

🔒 مناسب برای بسیاری از پروتالها

🔒 راهکار ۲: همیشه طول پیام را به عنوان قطعه اول به تابع CBC-MAC

بدهیم

🔒 راهکار ۳: قطعه آخر (C_N) را یک مرتبه مجددا رمز کنیم

🔒 اثبات شده است که هر سه راهکار امن هستند

سوالات متداول در مورد MAC

آیا MAC همانند امضا غیر قابل انکار است؟

خیر

چون گیرنده نیز می تواند برچسب را تولید کند

امضا با یک زوج کلید عمومی/خصوصی فراهم می شود

ولی کلید MAC یک کلید مشترک سری است

بر خلاف امضا، دو طرف قادر به ایجاد MAC هستند

معایب تولید MAC با رمزنگاری

❗ ایراد اصلی: کارایی پایین

❗ الگوریتم های بسیار سریع تری برای تولید MAC وجود دارد

▪ مثال: بکارگیری توابع درهم ساز

فهرست مطالب

- ❏ مفاهیم اولیه
- ❏ رمزگذاری پیام و کدهای تشخیص خطا
- ❏ کدهای تصدیق صحت پیام
- ❏ اصول توابع درهم ساز
- ❏ توابع درهم ساز مهم
- ❏ HMAC

توابع درهم ساز (Hash)

- 🔒 تابع یک طرفه
- 🔒 طول ورودی دلخواه
- 🔒 طول خروجی ثابت (نگاشت از فضای بزرگتر به فضای کوچکتر)
- 🔒 در حالت کلی، کلیدی در کار نیست!
- 🔒 بر خلاف MAC و رمزنگاری

امنیت توابع درهم ساز: ایده کلی

🔒 نگاشت پیام‌های طولانی به رشته‌های کوتاه به نحوی که:

🔒 یافتن پیام‌های متفاوتی که به یک رشته یکسان نگاشته می‌شوند دشوار باشد

🔒 به این رشته، عصاره یا چکیده پیام (Message Digest) می‌گوییم

نیازمندیهای امنیتی توابع درهم ساز

❏ دشواری یافتن پیش نگاره (Preimage Resistance)

❏ نام دیگر: یک طرفه بودن (One-wayness)

❏ دشواری یافتن پیش نگاره دوم (2nd Preimage Resistance)

❏ نام دیگر: مقاومت ضعیف در برابر تصادم (Weak Collision Resistance)



❏ مقاومت در برابر تصادم (Collision Resistance)

❏ نام دیگر: مقاومت قوی در برابر تصادم (Strong Collision Resistance)

تعریف نیازمندیهای امنیتی توابع درهم‌ساز

🔒 دشواری یافتن پیش‌نگاره:

🔒 فقط با داشتن $H(x)$ (برای یک x تصادفی)، یافتن y به طوری که $H(x)=H(y)$ از لحاظ محاسباتی ناممکن باشد

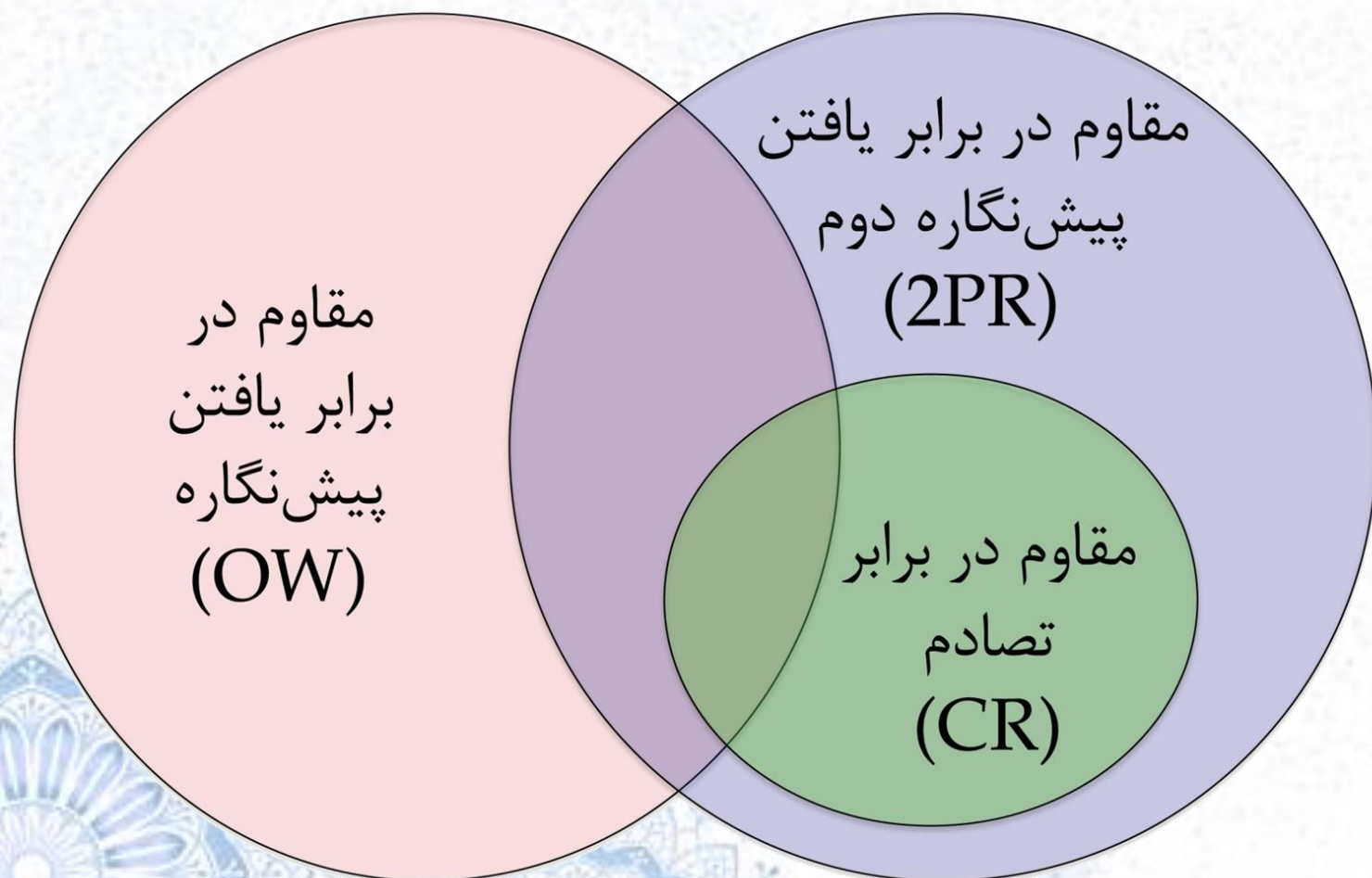
🔒 دشواری یافتن پیش‌نگاره دوم:

🔒 با داشتن x (و در نتیجه $H(x)$)، یافتن y به طوری که $H(x)=H(y)$ و $y \neq x$ از لحاظ محاسباتی ناممکن باشد

🔒 مقاومت در برابر تصادم:

🔒 یافتن $y \neq x$ به طوری که $H(x)=H(y)$ از لحاظ محاسباتی ناممکن باشد

نمودار ون انواع توابع درهم ساز



CR نتیجه می دهد 2PR

اثبات با برهان خلف:

فرض کنیم تابع H خاصیت CR دارد ولی خاصیت 2PR ندارد (فرض خلف)

یک x تصادفی تولید کنید

■ یافتن $y \neq x$ به طوری که $H(x)=H(y)$ از لحاظ محاسباتی ممکن است (چون 2PR نداریم)

یک زوج x و y با مقدار درهم سازی یکسان پیدا کردیم

پس خاصیت CR وجود ندارد

تناقض!

CR الزاماً نتیجه نمی دهد OW

مثال نقض: تابع $H(x)=x$

این تابع CR (و 2PR) است چون هیچ تصادفی ندارد

این تابع OW نیست چون پیشنگاره هر مقداری را می توان یافت

نتیجه:

2PR الزاماً OW را نتیجه نمی دهد

توابع فشرده ساز:

اگر برد H نسبت به دامنه آن خیلی کوچک باشد:

■ CR نتیجه می دهد OW

همه توابع درهم ساز در عمل چنین هستند

OW الزاماً نتیجه نمی دهد 2PR

🔒 مثال نقض:

🔒 فرض کنید $n=pq$ حاصلضرب دو عدد اول باشد

🔒 می توان ثابت کرد که $f(x)=x^2 \bmod n$ تابعی یک طرفه است

🔒 با فرض دشواری تجزیه n

🔒 با این حال خاصیت 2PR (و در نتیجه CR) را ندارد

🔒 هر x با مقدار $-x$ تصادم دارد

🔒 دو مقدار متفاوت با مقدار درهم سازی یکسان

حمله آزمون جامع به توابع درهم ساز

🔒 تابع درهم ساز زیر را در نظر بگیرید:

🔒 ورودی به طول دلخواه و خروجی به طول n

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$

🔒 پیچیدگی یافتن تصادم:

🔒 به طور متوسط پس از امتحان حدودا $1.25 \times 2^{n/2}$ ورودی با احتمال ۵۰٪ یک تصادم پیدا می شود

🔒 علت:

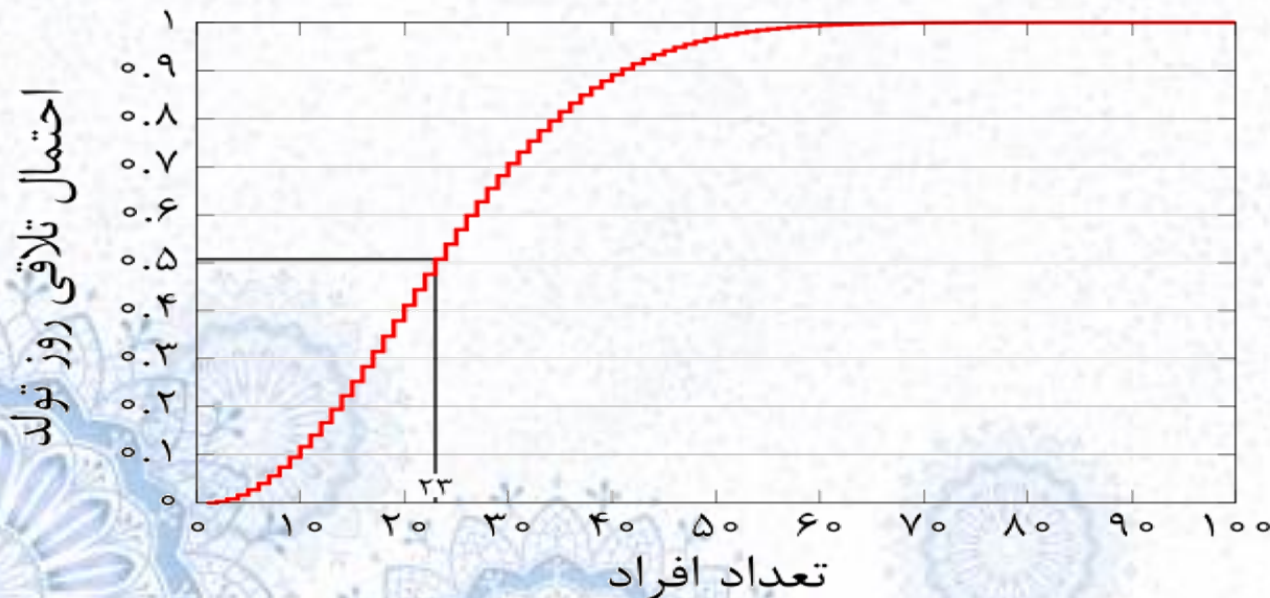
🔒 تناقض روز تولد

تناقض روز تولد

🔒 در میان ۲۳ نفر،

🔒 احتمال یافتن دو نفر که در یک روز از سال متولد شده‌اند بیش از ۵۰٪ است

$$1,25 \times \sqrt{365} \approx 23$$



نمونه ای از حمله روز تولد

🔒 Tom و Bob دو کاندیدای استخدام هیئت علمی هستند

🔒 Tom می‌داند که رئیس دانشکده نسبت به وی نظر مثبتی دارد

🔒 از وی می‌خواهد که برایش توصیه نامه‌ای بنویسد، و پس از امضا به دفتر رئیس دانشگاه بفرستد

🔒 فرض کنیم برای امضا، چکیده ۶۴ بیتی متن نامه تهیه شده و این چکیده امضا می‌شود

🔒 منشی رئیس دانشکده که با Tom خصومت دارد، دو نامه جداگانه تهیه می‌کند...

🔒 یک نامه با نظر مثبت

🔒 و یک نامه با نظر منفی

🔒 هر کدام با ۳۲ انتخاب

نامه اول (شامل ۳۲ انتخاب دو تایی)

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.

He is also a [highly | greatly] [respected | admired] [teacher | educator]. His students give his [classes | courses] [rave | spectacular] reviews. He is [our | the Department's] [most popular | best-loved] [teacher | instructor].

[In addition | Additionally] Prof. Wilson is a [gifted | effective] fund raiser. His [grants | contracts] have brought a [large | substantial] amount of money into [the | our] Department. [This money has | These funds have] [enabled | permitted] us to [pursue | carry out] many [special | important] programs, [such as | for example] your State 2016 program. Without these funds we would [be unable | not be able] to continue this program, which is so [important | essential] to both of us. I strongly urge you to grant him tenure.

نامه دوم (شامل ۳۲ انتخاب دو تایی)

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] Tom for [about | almost] six years. He is a [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.

Furthermore, he is not a [respected | admired] [teacher | educator]. His students give his [classes | courses] [poor | bad] reviews. He is [our | the Department's] least popular [teacher | instructor], known [mostly | primarily] within [the | our] Department for his [tendency | propensity] to [ridicule | embarrass] students [foolish | imprudent] enough to ask questions in his classes.

[In addition | Additionally] Tom is a [poor | marginal] fund raiser. His [grants | contracts] have brought only a [meager | insignificant] amount of money into [the | our] Department. Unless new [money is | funds are] quickly located, we may have to cancel some essential programs, such as your State 2016 program. Unfortunately, under these [conditions | circumstances] I cannot in good [conscience | faith] recommend him to you for [tenure | a permanent position].

چگونگی جعل...

🔒 حال منشی به کمک کامپیوتر:

🔒 جدولی از هر یک از 2^{32} انتخاب ممکن برای نامه اول، به همراه چکیده متناظر به عنوان **کلید جدول** تشکیل می‌دهد

🔒 چکیده هر یک از 2^{32} انتخاب ممکن برای نامه دوم را در جدول جستجو میکند، تا به اولین تساوی برسد

🔒 در صورتی که چکیده **L1** (یکی از انتخابها برای نامه اول) و

چکیده L2 (یکی از انتخابها برای نامه دوم) مساوی شوند..

🔒 منشی **L1** را به امضای رئیس دانشکده میرساند؛

🔒 ولی **L2** را به همراه امضا ارسال می‌کند

ساختار مرکب – دَمگارد (MD)

- مورد استفاده در بسیاری از توابع درهم‌ساز معروف
- اعمال مکرر یک تابع فشرده‌ساز به یک رشته با طول ثابت
- اگر تابع فشرده‌ساز CR باشد، تابع درهم‌ساز نیز CR خواهد بود



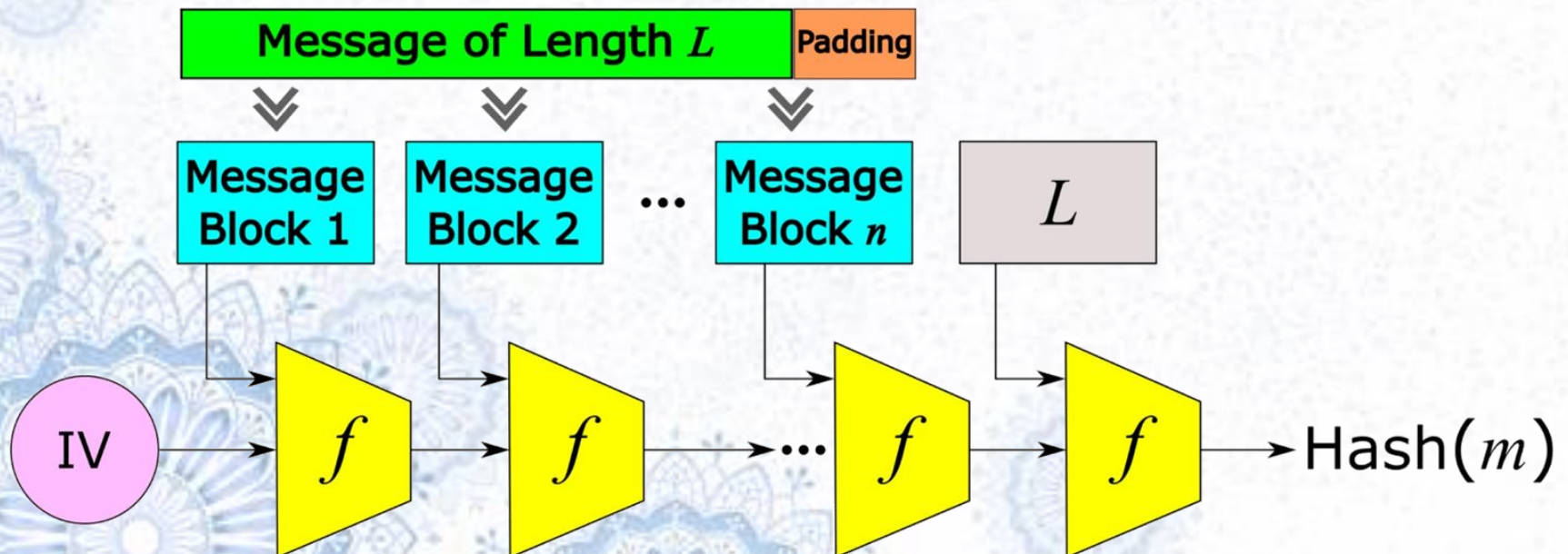
Ralph Merkle
(1952 –)



Ivan Bjerre
Damgård (1956 –)

ساختار درونی توابع درهم‌ساز

- 🔒 تابع فشرده ساز: f
- 🔒 این تابع باید CR باشد
- 🔒 IV: مقداری ثابت



ضعف MD در برابر حملات افزایش طول

❗ ساختار MD در برابر حملات افزایش طول آسیب پذیر است

❗ با داشتن $H(x)$ برای x نامعلوم به طول L ، می توان مقدار:

$$H(x \parallel \text{pad}(x) \parallel L \parallel y)$$

را برای y دلخواه به دست آورد

❗ حمله به Flickr در سال ۲۰۰۹ با این روش

❗ راه حل:

❗ طول پیام را به عنوان قطعه نخست به ساختار MD داد

❗ یا برای قطعه آخر از یک تابع فشرده ساز متفاوت استفاده کرد

فهرست مطالب

- ❏ مفاهیم اولیه
- ❏ رمزگذاری پیام و کدهای تشخیص خطا
- ❏ کدهای تصدیق صحت پیام
- ❏ اصول توابع درهم ساز
- ❏ توابع درهم ساز مهم
- ❏ HMAC

تابع MD5

MD5: Message Digest 5

طراحی در ۱۹۹۲ توسط رایوست، یکی از سه طراح RSA
استفاده گسترده در گذشت

امروزه SHA-1 و SHA-2 جایگزین آن شده‌اند

ویژگیها:

ساختار مرکل-دمگارد

پیام به قطعات ۵۱۲ بیتی تقسیم می‌شود

خروجی ۱۲۸ بیتی



Ronald Linn Rivest
(1947 –)

امنیت MD5

❏ مقاومت در برابر حمله روز تولد: 2^{64} گام

❏ امروزه امن محسوب نمی‌شود

❏ حملات کاراتری نیز به این الگوریتم پیدا شده است:

❏ بهترین حمله تصادم: سال ۲۰۱۳

❏ حمله در 2^{18} گام

❏ کمتر از ۱ ثانیه

❏ در سال ۲۰۱۲ ویروس Flame با سوء استفاده از این حمله به MD5 امضای دیجیتال مایکروسافت را جعل کرد

تابع SHA-1

SHA-1: Secure Hash Algorithm – 1

استاندارد NIST، سال ۱۹۹۵

ساختار مرکل-دمگارد

طول ورودی کوچکتر از 2^{64} بیت

طول خروجی ۱۶۰ بیت

امنیت SHA-1

- ❏ مقاوم در برابر تصادم با حمله روز تولد : 2^{80} گام
- ❏ بهترین حمله: در سال ۲۰۱۱ توسط Marc Stevens
- ❏ پیچیدگی حمله بین $2^{60/3}$ و $2^{65/3}$
- ❏ جایگزینی با گونه‌های امن‌تر
- ❏ خانواده SHA-2
- ❏ در حال استفاده گسترده در حال حاضر

توابع درهم ساز مهم: SHA-2

🔒 نسخه‌های زیر نیز علاوه بر SHA-1 استاندارد شده‌اند:

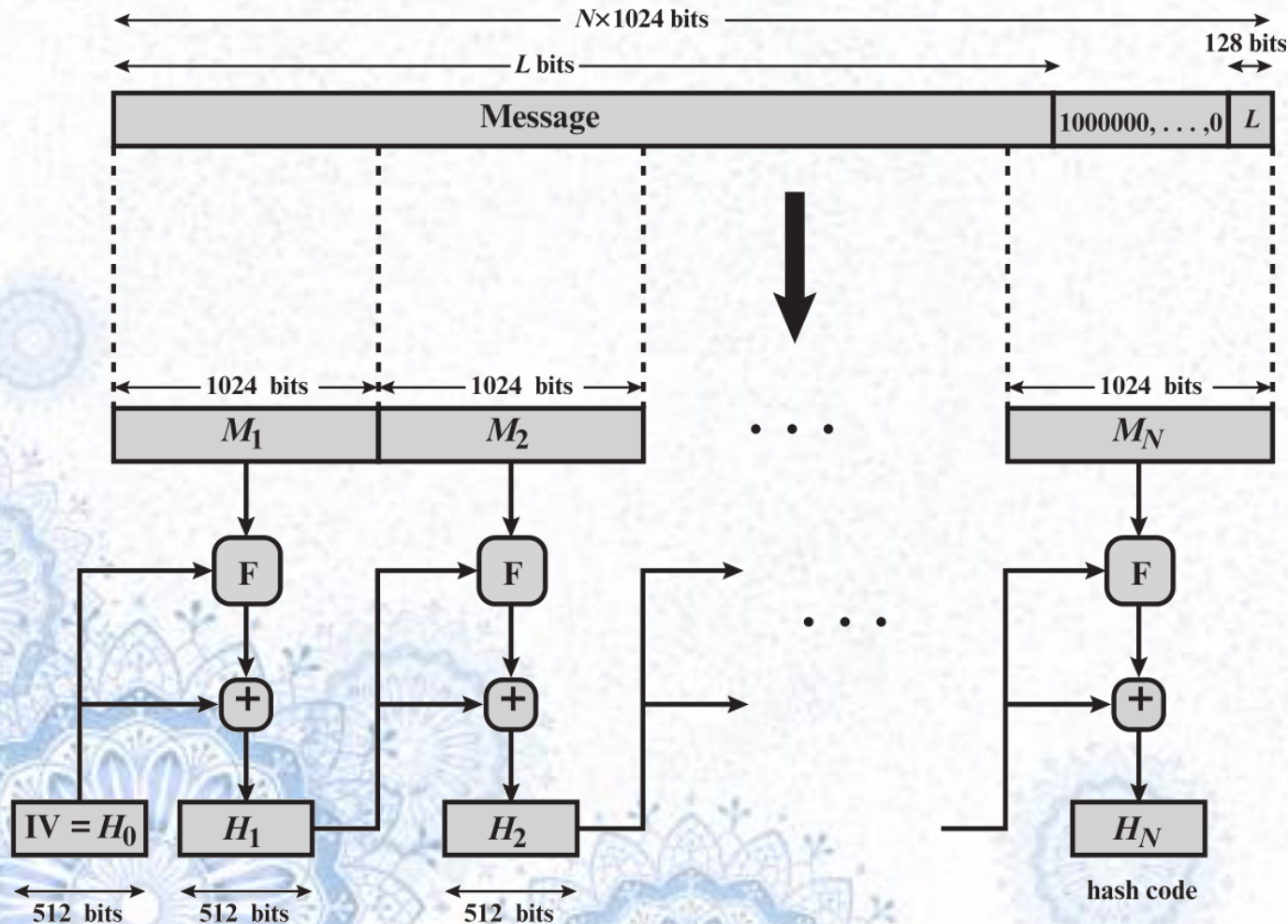
SHA-224، SHA-256، SHA-384 و SHA-512 🔒

معروف به خانواده SHA-2 هستند 🔒

از لحاظ ساختار و جزئیات مشابه SHA-1 هستند 🔒

Algorithm	Digest size	Block size	Message size	CR Security
SHA-1	160	512	$< 2^{64}$	80 bits
SHA-224	224	512	$< 2^{64}$	112 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

ساختار SHA-512



مشکلات

🔒 MD5، SHA-1 و تمامی توابع خانواده SHA-2 در برابر حملات افزایش طول آسیب پذیر هستند

🔒 اگر بخواهیم پیام m را با $H(K||m)$ تصدیق هویت کنیم،

🔒 مهاجم برای مقدار دلخواه m' می تواند به سادگی مقدار

$H(K||m||pad||L||m')$ را به دست آورد

🔒 ضعف در برابر حمله تصادم جزئی از پیام برای همه توابع تکراری

🔒 با یافتن تصادم در تابع فشرده ساز

راهکار: استاندارد SHA-3

❏ ساختار غیر مرکب-دمگارد

❏ دارای ساختار توابع اسفنجی

❏ مقاوم در برابر حمله افزایش طول

❏ استاندارد شده در ۲۰۱۵

❏ به عنوان مکمل و (نه جایگزین) SHA-2

❏ پشتیبانی از طول خروجی‌های ۲۲۴، ۲۵۶، ۳۸۴، ۵۱۲

فهرست مطالب

- ❏ مفاهیم اولیه
- ❏ رمزگذاری پیام و کدهای تشخیص خطا
- ❏ کدهای تصدیق صحت پیام
- ❏ اصول توابع درهم ساز
- ❏ توابع درهم ساز مهم
- HMAC** ❏

کد تصدیق اصالت HMAC

🔒 HMAC روش است برای ترکیب کلید مخفی با الگوریتم‌های درهم‌ساز فعلی
🔒 با توجه به اینکه $H(K||m)$ یا $H(m||K)$ برای توابع درهم‌ساز فعلی امن
نیستند

🔒 راهکار :

🔒 دو مرحله استفاده از H

🔒 یعنی $H(K_2||H(K_1||m))$

🔒 بهتر است کلیدهای درونی و بیرونی متفاوت باشند

🔒 امروزه HMAC به طور گسترده استفاده می‌شود

🔒 به طور مثال در SSL و IPsec

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

❏ لایه دوم Hash معادل ۱ یا ۲ تابع فشرده‌سازی

❏ استفاده ساده از کلید

نماد گذاری الگوریتم HMAC

H : تابع درهم سازی به کار گرفته شده (با خروجی n بیتی)

M : پیام ورودی (با قطعه‌های b بیتی)

K : کلید مخفی

طول آن باید بیشتر از n باشد

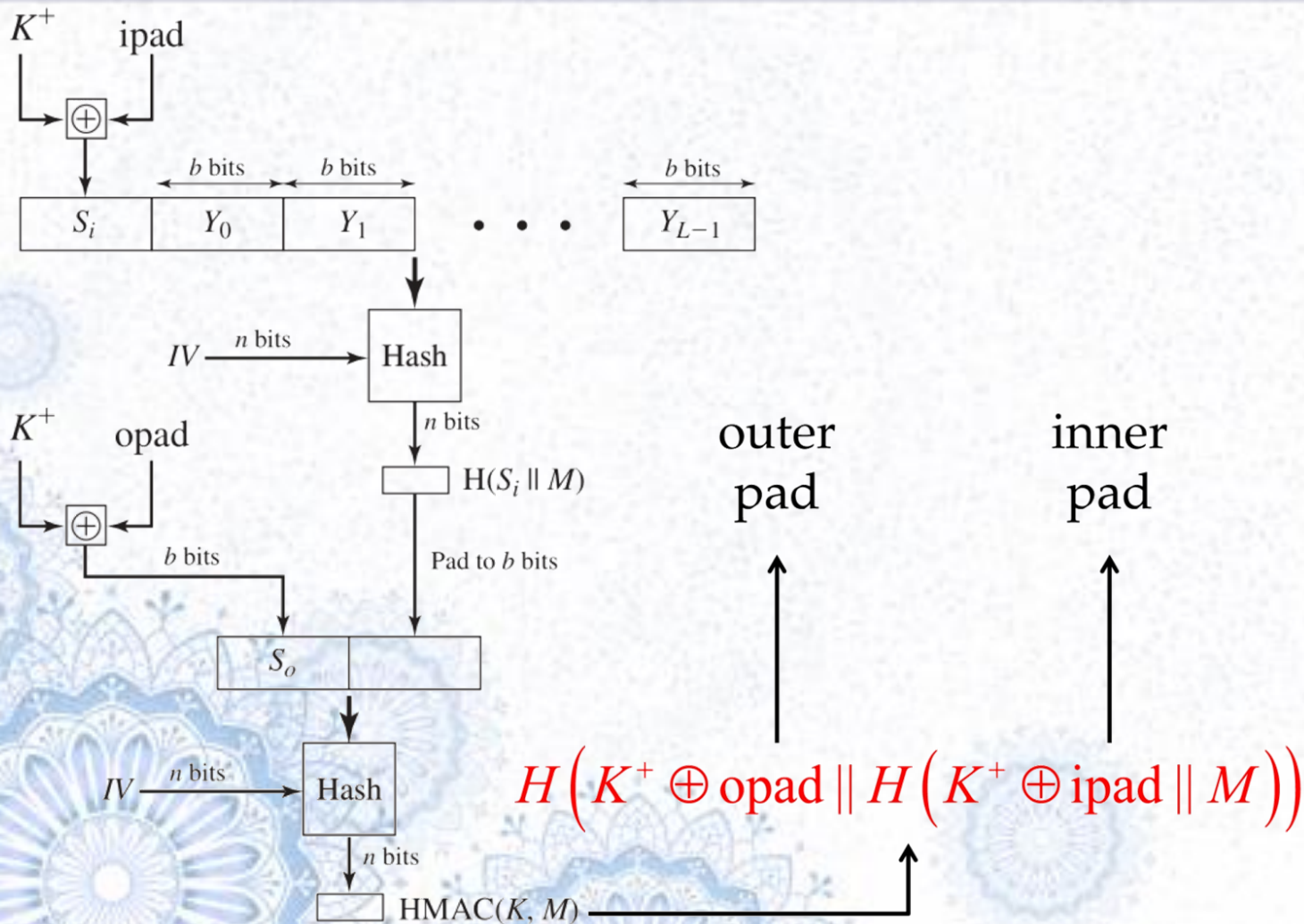
اگر طول کلید بیشتر از b بود $H(k)$ به جای آن استفاده می‌شود

K^+ : کلید مخفی که از سمت چپ با صفر pad شده است تا به طول b برسد

ipad : رشته b بیتی حاصل از تکرار بایت $0x36$ (به تعداد $b/8$)

opad : رشته b بیتی حاصل از تکرار بایت $0x5c$ (به تعداد $b/8$)

نحوه محاسبه HMAC



امنیت HMAC

ارتباط دقیق بین امنیت HMAC با امنیت تابع درهم ساز اثبات شده است

حمله به HMAC:

حمله آزمون جامع بر روی کلید (میزان مقاومت بسته به طول کلید)

حمله روز تولد: با توجه به نداشتن کلید، نیازمند مشاهده تعداد زیادی پیام و HMAC آنها با کلید یکسان

مقاومت HMAC در برابر حمله روز تولد از تابع درهم ساز به کار گرفته شده، بیشتر است