دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکدهٔ مهندسی کامپیوتر و فنآوری اطلاعات

مبانی امنیت اطلاعات

(ترم بهمن ۹۸)

Hash Functions

نستوه طاهری جوان

nastooh@aut.ac.ir

توابع Hash

- ✓ چند خصوصیت اصلی توابع درهم ساز
  - عموماً یک ورودی و یک خرودی دارند.
    - به خروجی تابع درهم ساز Message Digest یا Hash code گویند.
  - طول ورودی متغییر، ولی طول خروجی ثابت است.
  - احتمال وقوع تصادم در آنها نزدیک به صفر است.
    - تصادم: دو ورودی متفاوت با یک خروجی یکسان
  - یکطرفه هستند.
    - با داشتن یک خروجی، نمی توان ورودی آن را حدس زد.
  - با اعمال تغییر در ورودی (حتی به اندازه یک بیت) خروجی کاملا متفاوتی تولید می شود.

nastooh@aut.ac.ir

توابع Hash

- ✓ چند خصوصیت اصلی توابع درهم ساز
  - بحث در مورد خصوصیت دوم و سوم!!!!

nastooh@aut.ac.ir

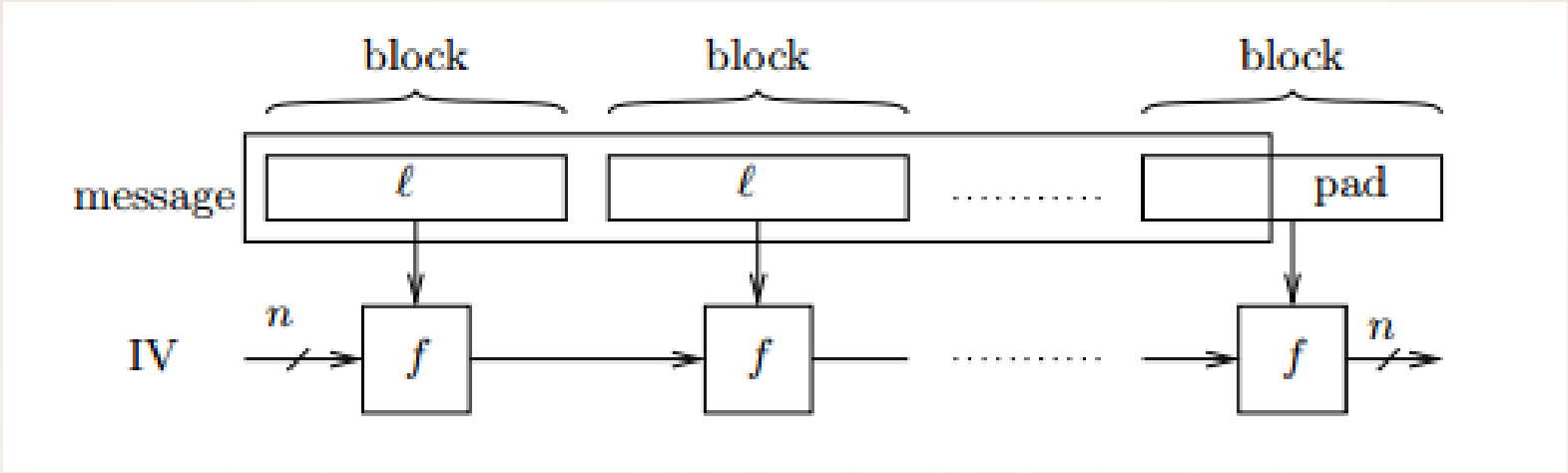
توابع Hash

- ✓ کاربردهای توابع درهم ساز
  - بررسی جامعیت (صحت) اطلاعات
  - استفاده به عنوان اندیس دسترسی به داده
  - و و و
  - امضای دیجیتال

nastooh@aut.ac.ir

الگوی مِرکِل-دَمگارد

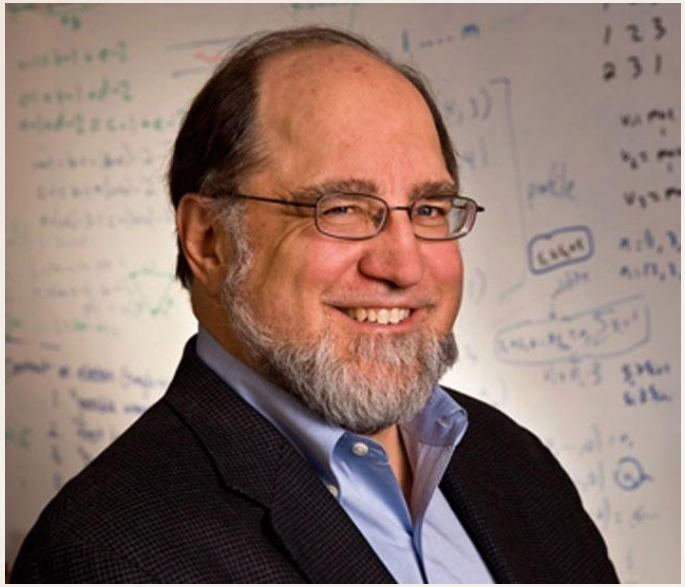
- ورودی با طول متغییر
- خروجی با طول ثابت
- ساختار بلاکی
- f، تابع فشرده ساز



nastooh@aut.ac.ir

MD5

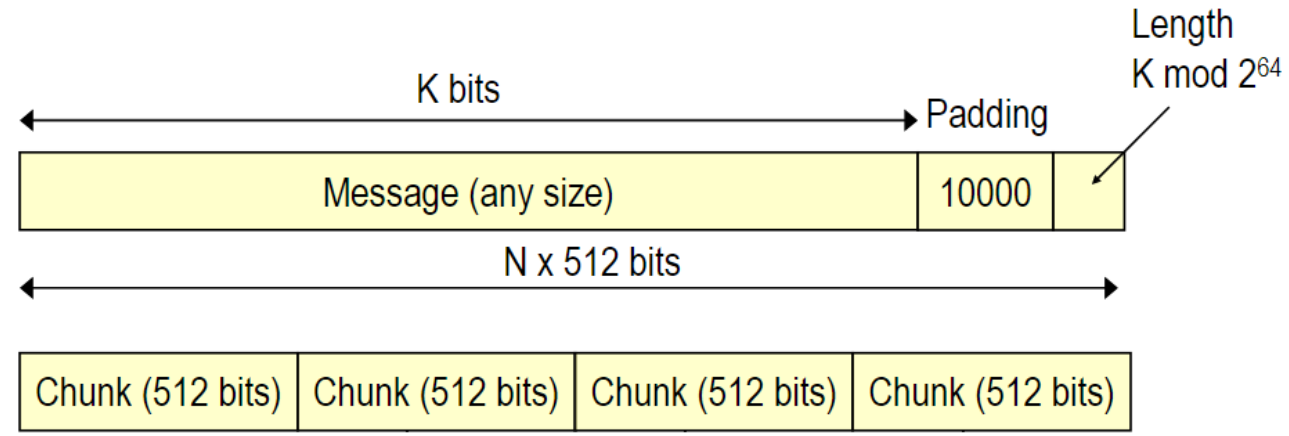
- تابع در هم سازی MD5 (Message Digest v5)
  - مبتنی بر ساختار مِرکل-دَمگارد
  - پیشنهاد شده توسط رونالد ری‌وست (حرف R در RSA)
  - ارائه در سال ۱۹۹۱ بعد از ۴ نسخهٔ پیشین.
  - طول خروجی: ۱۲۸ بیت.



nastooh@aut.ac.ir

MD5

- مرحلهٔ اول: افزودن دادهٔ اضافه (Padding)
  - هدف: تنظیم طول داده اصلی
  - الگو: تعدادی بلاک ۵۱۲ بیتی، و آخرین بلاک ۴۴۸ بیت
  - روال: افزودن یک بیتِ ۱، به همراه تعداد مناسب بیتِ ۰ به انتهای پیام
- مرحلهٔ دوم: افزودن طول دادهٔ اصلی
  - طول دادهٔ اصلی در قالب یک عدد ۶۴ بیتی به انتهای پیام افزوده می شود.
  - نتیجه: همهٔ بلاک ها ۵۱۲ بیتی می شوند.



nastooh@aut.ac.ir

MD5

- مرحلهٔ سوم: مقدار دهی های اولیه
  - تعدادی متغییر و آرایه با مقادیر ثابت مقداردهی اولیه می شوند.
    - در ادامه معرفی خواهند شد.

- مرحلهٔ چهارم: پردازش پیام در بلوک های ۵۱۲ بیتی
  - هر بلوک در ۴ دورِ ۱۶ تایی پردازش می شوند. (جمعا ۶۴ دور)
  - عملیات هر تکرار شامل AND، OR، NOT و XOR بهمراه جایگشت است.

خروجی هر بلاک، با بلاک بعدی جمع (بیتی) خواهد شد تا نتیجهٔ نهایی حاصل گردد.

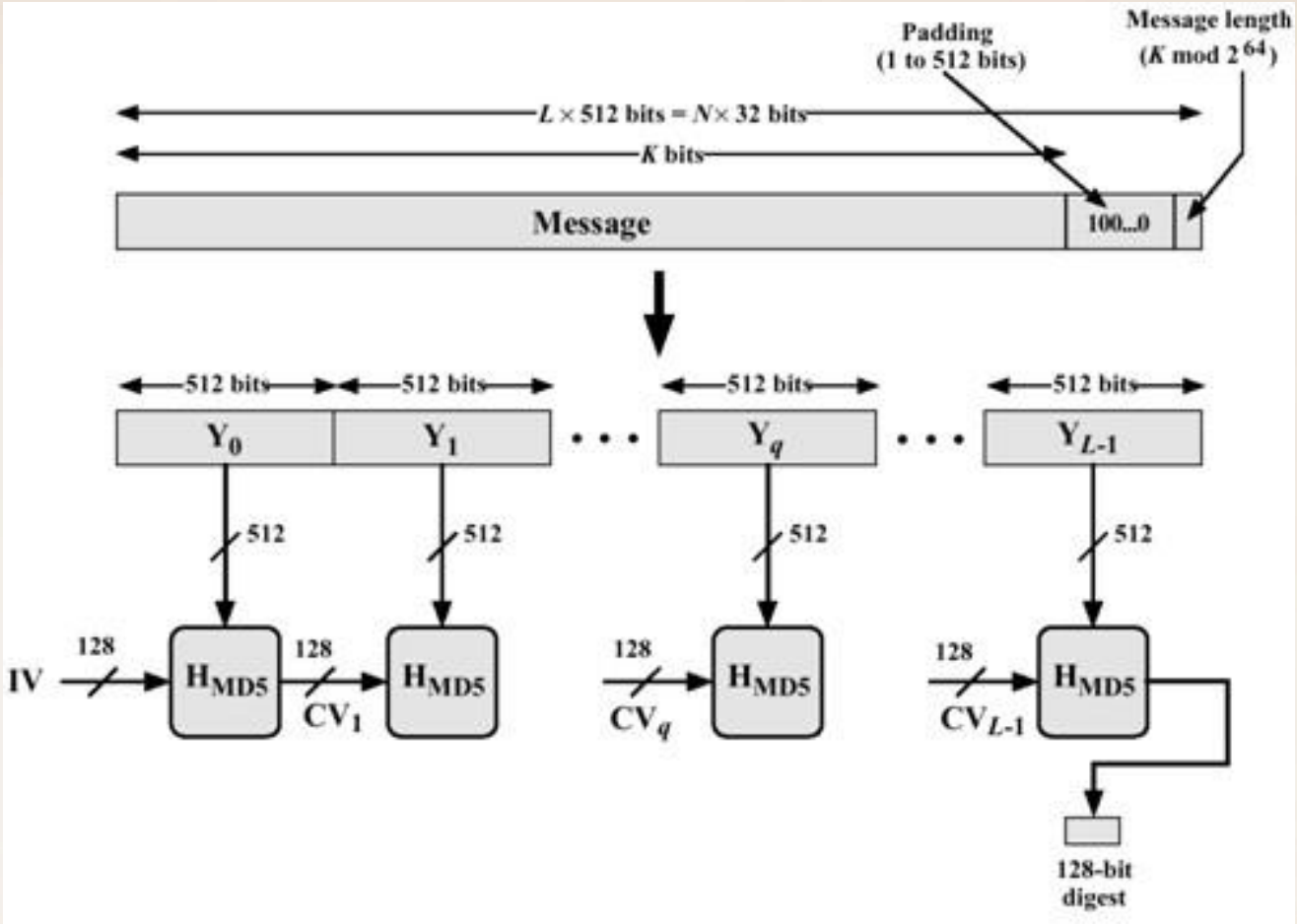
nastooh@aut.ac.ir

MD5

- ✓ شمای کلی الگوریتم

IV:
Initial Value

CV:
Chaining Variable



nastooh@aut.ac.ir

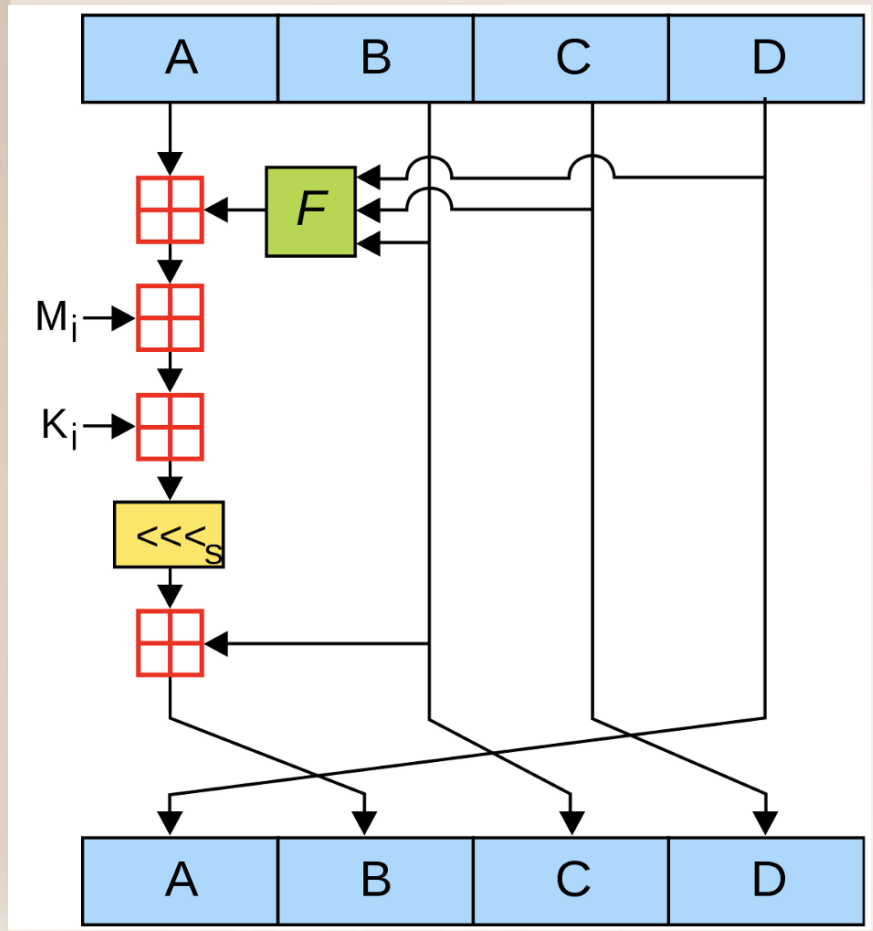
MD5

- جزئیات پردازش هر دور
  - دادهٔ هر بلوک در شانزده کلمهٔ ۳۲ بیتی و در آرایهٔ w[0, . . . , 15] ذخیره می شود.
  - در هر مرحله از ۴ متغیر ۳۲ بیتی با نامهای A، B، C و D استفاده می شود.
    - در ابتدای هر مرحله (به ازای هر بلوک) چهار متغیر A، B، C و D با مقادیر مشخصی مقداردهی اولیه می شوند.
      - « این مقدار اولیه در واقع مقدار هَش بلوکهای قبلی است.
  - در هر مرحله این چهار متغیر طِی ۶۴ دور محاسبه می شوند.
    - محاسبات بر اساس دادهٔ ورودی بلوک و تعدادی ثابت صورت می گیرد.
    - این ۶۴ دور، در واقع ۴ دورِ ۱۶ تایی است.

nastooh@aut.ac.ir

MD5

- جزئیات پردازش هر دور (۶۴ گانه)

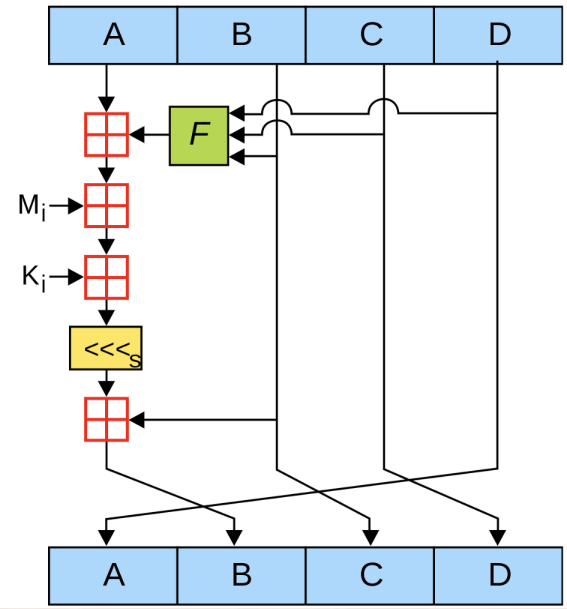


- آرایه M بر اساس داده ورودی (Wها) در هر دور
- مقدار دهی می شود.
- آرایهٔ K یک آرایهٔ ثابت ۶۴ عنصری است.
- تابع F در هر مرحله، هر ۱۶ دور تغییر می کند. یعنی چها رحالت مختلف دارد.
- شیفت چرخشی هر در هر دور ۶۴ گانه، بر اساس یک آرایهٔ ثابت (r) شیف می دهد.
- در هر دور، مقدار B به جای C، مقدار C به جای D و مقدار D به جای A قرار می گیرد. و مقدار B از نو محاسبه می شود.

nastooh@aut.ac.ir

MD5

- جزئیات پردازش هر دور



- فرمول تابع F به ازای ۱۶ دورِ اول (۰ تا ۱۵)

F=(B and C) or ((not B) and D)

- فرمول تابع F به ازای ۱۶ دور دوم (۱۶ تا ۳۱)

F=(D and B) or ((not D) and C)

- فرمول تابع F به ازای ۱۶ دور سوم (۳۲ تا ۴۷)

F=B xor c xor d

- فرمول تابع F به ازای ۱۶ دور چهارم (۴۷ تا ۶۳)

F=C xor (B or (not D))

nastooh@aut.ac.ir

MD5

- روال کار

- متغیرهای A، B، C و D در ابتدای هر مرحله (بلوک) با مقدار هَش بلوکهای تا قبل از بلوک جاری مقدار دهی می شوند.

- در طول کل فرآیند چهار متغیر H0، H1، H2 و H3 وجود دارند که با مقدار اولیهٔ پیش فرضی مقدار دهی می شوند و در طول فرآیند، در هر دور به روز می شوند.

- در انتهای هر بلوک، مقادیر A، B، C و D با متغیرهای H0، H1، H2 و H3 جمع می شوند. (بدون رقم نقلی)
  - خروجی هَش کل الگوریتم در H0، H1، H2 و H3 ذخیره شده است.

nastooh@aut.ac.ir

MD5

- مقادیر اولیه

- شیف های هر دورِ ۶۴گانه بر اساس مقادیر ثابتی که در یک آرایه 64 عنصری با عنوان r[0, …, 63] ذخیره شده اند، صورت می گیرد. مقادیر آن بین ۴ تا ۲۳ هستند.

- آرایهٔ ۶۴ عنصری K[0, …, 63] یک آرایهٔ ثابت با مقادیر رندوم است. (عملا در حین اجرا به طور ثابت و با فرمولی مبتنی بر سینوس تولید می شود.)

- هر یک از مقادیر ۶۴ گانهٔ M که در عملیات جمع های هر دور شرکت می کند، بر اساس فرمول ثاتبی برابر با یکی از مقادیر ۱۶ گانهٔ w (دادهٔ اصلی رودی بلوک) مقدار دهی می شوند.

nastooh@aut.ac.ir

MD5

- مقادیر اولیه
  - چهار متغیر H0، H1، H2 و H3:

H0= 67452301, H1= EFCDAB89,
H2=98BADCFE, H3= 10325476,

- آرایهٔ r برای شیف های ۶۴ دور:

```
r[ 0..15] := { 7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22 }
r[16..31] := { 5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20 }
r[32..47] := { 4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23 }
r[48..63] := { 6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21 }
```

- آرایهٔ ۱۶ عنصری w نیز با دادهٔ اصلی بلاک (۵۱۲ بیتی) مقدار دهی می شود

nastooh@aut.ac.ir

MD5

- مقادیر اولیه
  - آرایه ۶۴ عنصری K (**یا** در حین اجرا بر اساس فرمول، **یا** با مقدار دهی اولیه)

```
for i from 0 to 63 do
    K[i] := floor(2^32 × abs (sin(i + 1)))
end for
// (Or just use the following precomputed table):
K[ 0.. 3] := { 0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee }
K[ 4.. 7] := { 0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501 }
K[ 8..11] := { 0x698098d8, 0x8b44f7af, 0xffff5bb1, 0x895cd7be }
K[12..15] := { 0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821 }
K[16..19] := { 0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa }
K[20..23] := { 0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fbc8 }
K[24..27] := { 0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed }
K[28..31] := { 0xa9e3e905, 0xfcefa3f8, 0x676f02d9, 0x8d2a4c8a }
K[32..35] := { 0xfffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c }
K[36..39] := { 0xa4beea44, 0x4bdecfa9, 0xf6bb4b60, 0xbebfbc70 }
K[40..43] := { 0x289b7ec6, 0xeaa127fa, 0xd4ef3085, 0x04881d05 }
K[44..47] := { 0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665 }
K[48..51] := { 0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039 }
K[52..55] := { 0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1 }
K[56..59] := { 0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1 }
K[60..63] := { 0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391 }
```


nastooh@aut.ac.ir

MD5

```
for each 512-bit chunk
    var int A := a0
    var int B := b0
    var int C := c0
    var int D := d0
    for i from 0 to 63 do
        var int F, g
        if 0 ≤ i ≤ 15 then
            F := (B and C) or ((not B) and D)
            M[i] := w[i]
        else if 16 ≤ i ≤ 31 then
            F := (D and B) or ((not D) and C)
            M[i] := w[(5×i + 1) mod 16]
        else if 32 ≤ i ≤ 47 then
            F := B xor C xor D
            M[i] := w[(3×i + 5) mod 16]
        else if 48 ≤ i ≤ 63 then
            F := C xor (B or (not D))
            M[i] := w[(7×i) mod 16]
        F := F + A + K[i] + M[i]
        A := D
        D := C
        C := B
        B := B + leftrotate(F, r[i])
    end for
    H1 := H1 + A
    H2 := H2 + B
    H3 := H3 + C
    H4 := H4 + D
end for
```

شبه کد مربوط به محاسبات هر بلاک ۵۱۲ بیتی

مقدار دهی های اولیه یک بار، انجام شده اند.

شامل:

متغییرهای H1، H2، H3 و H4

آرایه ۱۶ گانه w با دادهٔ اصلیِ بلاک

آرایه ۶۴ گانه K با مقادیر ثابت همیشگی

آرایهٔ ۶۴ گانهٔ r با ثابت همیشگی (برای شیفت ها)

nastooh@aut.ac.ir

MD5

✓ استفاده از MD5

- MD5 از سال ۱۹۹۱ به مدت ۱۵ سال در بسیاری از کاربردها استفاده شد.
  - مانند نگهداری کلمات عبور در کاربردهای مختلف
  - حفظ سلامت فایل ها در لینوکس
  - و ...

✓ استحکام MD5

- در سال ۲۰۰۶ الگوریتمی پیدا شد که می توانست دو پیام متفاوت با هَش یکسان و در مدت زمان یک دقیقه بر روی یک کامپیوتر کاملا عادی پیدا کند! ☹
- کماکان در بسیاری از کاربردها از MD5 استفاده می شود.
- بعدها نسخهٔ MD6 نیز ارائه شد.

nastooh@aut.ac.ir

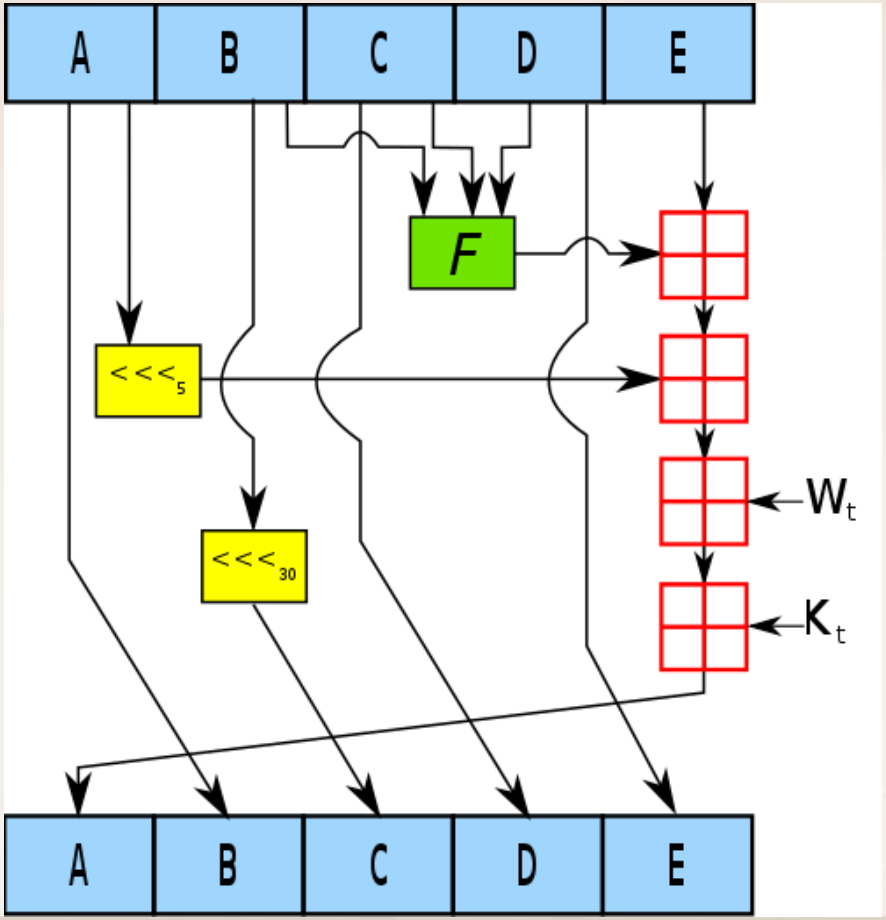
SHA1

- الگوریتم SHA1 (Secure Hash Algorithm v1)
  - در سال ۱۹۹۳ توسط NIST استاندارد و ارائه شد.
  - مبتنی بر ساختار مِرکل-دَمگارد بنا شده است.
  - در بسیاری از کاربردها مانند IPsec و SSL استفاده شده است.
  - خروجی آن ۱۶۰ بیتی است.
  - داده ها را در بلوک های ۵۱۲ بیتی پردازش می کند.
  - هر مرحلهٔ آن دارای ۸۰ دور است.

nastooh@aut.ac.ir

SHA1

- شمای کلی تابع اصلی الگوریتم



nastooh@aut.ac.ir

SHA1

✓ استحکام SHA1

- عملا در سال ۲۰۰۶ و با انتشار اخبار مربوط به کشف تصادم به آخر خطر رسید.
  - هنوز به طور گسترده و درادامهٔ پیاده سازی های قدیمی، در حال استفاده است!!
- ارائهٔ ردهٔ SHA2
  - در چهار نسخه: SHA224، SHA256، SHA384 و SHA512

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

- ارئهٔ SHA3 در سال ۲۰۱۵
  - برگزاری مسابقه برای ارائهٔ طرح، شبیه به AES
  - مانند SHA2 دارای چندین استاندارد با طولهای خلاصه پیام متفاوت...

nastooh@aut.ac.ir

توابع دیگر درهم ساز

- RIPEMD
- BLAKE
- Whirpool
- Streebog
- Kangaroo Twelve
- GOST
- Tiger
- RadioGatún
- and . . .

حملهٔ روز تولد

- ✓ کمی تامل و تحلیل پیرامون مسالهٔ تصادم

- **ابتدا یک سوال**:

  فرض کنید یک گردهمایی داریم!

  در این گردهمایی به طور متوسط چند نفر باید حضور داشته باشند تا احتمال آنکه جشنِ تولدِ حداقل دو نفر از آنها دقیقاً در یک روز باشد، از ۵۰٪ بیشتر باشد؟

حملهٔ روز تولد

- **پاسخ سوال قبل: ۲۳ نفر!**

نکته: سوال این نیست: چند نفر باید حضور داشته باشند تا جشن تولدشان با شما در یک روز باشد.

تحلیل ساده: با ۲۳ نفر، چند زوج مختلف می توانیم داشته باشیم؟

$$23 + 22 + 21 + \cdots + 1 = \frac{23 * 22}{2} = 253$$

حال هر یک از این زوج ها با احتمال $\frac{1}{365}$ در یک روز به دنیا آمده اند.

پس احتمال آنکه یکی از زوج ها در یک روز به دنیا آمده باشد، برابر $\frac{253}{365}$ است که از نیم بیشتر است.

nastooh@aut.ac.ir

حملهٔ روز تولد

- **حالت کلی:**
  - اگر به تعداد K نوع ویژگی مختلف داشته باشیم، تعداد گروه انتخابی (k) چند باشد تا با احتمال بالای ۵۰٪ دو عدد از این گروه دارای یک ویژگی یکسان باشند:

$$k \approx \sqrt{2ln2}\sqrt{K}$$

- مثال: در مثال روز تولد، K=365 بود، که k برابر با ۲۲.۵ شد.
- مثال: اگر در یک دریا ۵۲۵.۶۰۰ ماهی مختلف وجود داشته باشد، چند ماهی باید گرفته شود تا با احتمال بالای ۵۰ درصد، دوماهی یکسان صید شوند؟

K=525.600 و k برابر با ۸۵۴ می شود.

خب، متوجه شدیم، که چه؟

nastooh@aut.ac.ir

حملهٔ روز تولد

- حملهٔ روز تولد در مسالهٔ احتمال یافتن تصادم در یک تابع درهم ساز اهمیت دارد.

سوال: اگر چکیدهٔ یک تابع هش، n بیتی باشد، به طور متوسط چه تعداد پیام مختلف را باید چکیده کنیم تا احتمال وقوع تصادم در چکیده ها، بیش از ۵۰٪ باشد؟

حملهٔ روز تولد

- حملهٔ روز تولد در مسالهٔ احتمال یافتن تصادم در یک تابع درهم ساز اهمیت دارد.

سوال: اگر چکیدهٔ یک تابع هش، n بیتی باشد، به طور متوسط چه تعداد پیام مختلف را باید چکیده کنیم تا احتمال وقوع تصادم در چکیده ها، بیش از ۵۰٪ باشد؟

پاسخ: در این مساله فضای حالت ما 2^n است. یعنی داریم:

$$K = 2^n$$

$$k \approx \sqrt{2ln2}\sqrt{2^n} = 2^{n/2}$$

یعنی اگر طول چکیدهٔ پیامی ۳۲ بیت باشد (با ۴ میلیارد حالت مختلف)، اگر به طور متوسط ۶۵ هزار پیام مختلف را چکیده کنیم، با احتمال بالای ۵۰٪ تصادم خواهیم یافت.

nastooh@aut.ac.ir

حملهٔ روز تولد

این مساله صرفا برای تحلیل توابع هَش کاربرد دارد
و هنگام طراحی و آنالیز توابع هَش باید به آن دقت کرد.

- به عنوان مثال، برای کشف تصادف با احتمال بالای ۵۰٪:
  - در MD5 به طور متوسط 2^{64} تلاش
  - در SHA1 به طور متوسط 2^{80} تلاش
  - در SHA512 به صور متوسط 2^{256} تلاش

  نیاز است.

منابع

[1] William Stallings, “Cryptography and Network Security,” 7th ed.

nastooh@aut.ac.ir

پایان

nastooh@aut.ac.ir