

عنوان ارائه:

**بررسی الگوریتم‌های درهم‌سازی شکسته شده**

**A Review on Broken Hashing Algorithms**

توسط: علیرضا صادقی نسب

استاد: دکتر وحید رافع

تاریخ ارائه: 1400/08/24

# فهرست مطالب

- مقدمه
- الگوریتم‌های شکسته شده
  - الگوریتم  $SHA - 0$
  - الگوریتم  $SHA - 1$
  - الگوریتم  $MD2$
  - الگوریتم  $MD4$
  - الگوریتم  $MD5$
  - الگوریتم  $RIPEMD$
  - الگوریتم  $HAVAL$
  - الگوریتم  $Snefru$

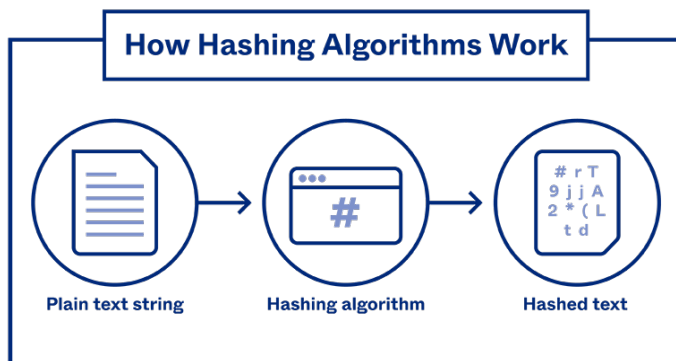
## مقدمه

### ■ الگوریتم‌های درهم‌سازی

★ یک الگوریتم درهم‌سازی یا Hash، تابعی است که داده ورودی را به یک خروجی رشته‌ای با طول ثابت تبدیل می‌کند. خروجی این تابع به طور کلی، از داده اصلی بسیار کوچکتر است

★ از الگوریتم‌های Hash برای کارهای مختلفی از جمله بررسی صحت یکپارچگی داده‌ها، ذخیره اطلاعات حساس و غیره استفاده می‌گردد

★ الگوریتم‌های Hash به گونه‌ای طراحی می‌شوند تا در برابر تصادم، مقاوم باشند. این بدان معنی است که احتمال تولید یک رشته یکسان برای دو داده متفاوت، باید بسیار کم باشد



## مقدمه

### ▪ حمله تصادم

★ در رمزنگاری، حمله تصادم به یک تابع درهم‌سازی سعی می‌کند دو ورودی مختلفی را پیدا کند که مقدار درهم‌شده هر دو، یکسان باشد.

★ به طور کلی، دو حمله تصادم وجود دارد:

☑ حمله تصادم: دو رشته مختلف  $m1$  و  $m2$  پیدا می‌شود به طوری که:

$$\text{hash}(m1) = \text{hash}(m2)$$

☑ حمله تصادم با پیشوند انتخاب شده: دو پیشوند متفاوت  $p1$  و  $p2$  و دو رشته  $m1$  و  $m2$  پیدا می‌شود به طوری که:

$$\text{hash}(p1 \parallel m1) = \text{hash}(p2 \parallel m2)$$



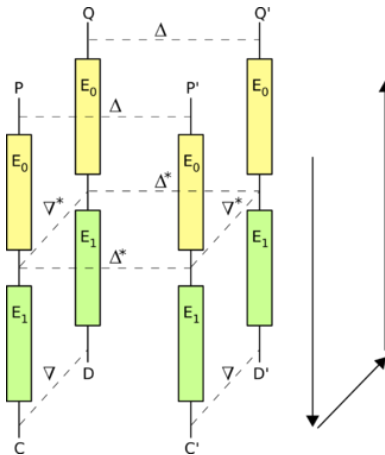
Concatenation Operator

## مقدمه

### ■ حمله بومرنگ

★ در رمزنگاری، حمله بومرنگ روشی برای تجزیه و تحلیل رمز (cryptanalysis) رمزهای بلوکی است. این روش مبتنی بر تحلیل رمز افتراقی (differential cryptanalysis) است. از این روش در ابتدا برای شکستن رمز COCONUT98 استفاده شده بود

★ همانطور که پیشتر اشاره شد، این حمله بر پایه تحلیل رمز افتراقی است. در این رویه، یک مهاجم از این ویژگی که چگونه تفاوت در ورودی یک متن ساده می‌تواند بر تفاوت حاصل در خروجی (متن رمز) تاثیر بگذارد، سوء استفاده می‌کند.



## مقدمه

### ■ حمله پیش تصویر

★ در رمزنگاری، حمله پیش تصویر سعی می‌کند یک رشته‌ای را پیدا کند که یک مقدار درهم شده خاصی را داشته باشد.

★ به طور کلی، دو نوع مقاومت در برابر حمله پیش تصویر داریم:

☑ مقاومت پیش تصویر: این مقاومت صرفاً مختص خروجی‌های از پیش مشخص شده است. تابع درهم‌سازی مقاوم، تابعی است که در آن از لحاظ محاسباتی، پیدا کردن ورودی‌ای که مقدار خروجی آن مشخص باشد، امکان‌پذیر نباشد.

☑ مقاومت پیش تصویر دوم: این مقاومت مختص ورودی از پیش مشخص شده است. تابع درهم‌سازی مقاوم، تابعی هست که در آن از لحاظ محاسباتی، پیدا کردن ورودی دیگری که مقدار درهم آن، خروجی مشابه‌ای را تولید کند، امکان‌پذیر نباشد.

# الگوریتم‌های شکسته شده

## ▪ الگوریتم SHA-0

\* الگوریتم در سال ۱۹۹۳ معرفی شد

\* تا سال ۱۹۹۷ به عنوان یک الگوریتم درهم‌سازی قوی تلقی می‌شد اما در سال ۱۹۹۸ تضعیف شد

\* در سال ۲۰۰۴ و ۲۰۰۵ چندین بار مورد حمله قرار گرفت. تمامی این حمله‌ها از جنس پیدا کردن تصادم بودند

\* در سال ۲۰۰۸ مورد حمله بومرنگ قرار گرفت و پیچیدگی پیدا کردن تصادم آن به قدری پایین آمد که تخمین زده شد یک کامپیوتر متوسط می‌تواند طی یک ساعت، این کار را انجام بدهد. پیچیدگی پیدا کردن تصادم‌ها در این روش،  $2^{33.6}$  می‌باشد

# الگوریتم‌های شکسته شده

## ▪ الگوریتم SHA-1

\* این الگوریتم در سال ۱۹۹۵ توسط سازمان NSA آمریکا معرفی و ثبت شد

\* استفاده‌های متعددی از SHA-1 در حوزه‌های مختلفی می‌شود از جمله، commit های Git همگی با این الگوریتم، درهم می‌شوند. البته استفاده از این الگوریتم در Git، به جهت امنیت نیست و صرفاً برای بررسی یکپارچگی و تشخیص دستکاری شدن فایل‌ها می‌باشد. از استفاده‌های مشهور دیگر این الگوریتم می‌توان به بکارگیری در پروتکل‌های TLS و SSL و IPsec نام برد

\* این الگوریتم تا سال ۲۰۰۳ به عنوان یک الگوریتم درهم‌سازی قوی تلقی می‌شد اما در سال ۲۰۰۴ شبههاتی از جهت تئوری در خصوص مقاومت آن مطرح شد تا اینکه در سال ۲۰۱۷ به صورت رسمی، یک تصادم برای آن پیدا شد. اطلاعات این تصادم در وبلاگ تخصصی امنیت گوگل منتشر شد



# الگوریتم‌های شکسته شده

<https://shattered.io>

The image displays two promotional cards for 'SHattered' and a terminal screenshot. The left card has a blue header with the text 'SHattered' and 'The first concrete collision attack against SHA-1' with the URL 'https://shattered.io'. Below the header are logos for 'CWI' and 'Google', and the names 'Marc Stevens' and 'Pierre Karpman'. The right card has a red header with the same text and logos, and the names 'Marc Stevens', 'Pierre Karpman', 'Elie Bursztein', 'Ange Albertini', and 'Yarik Markov'. The terminal screenshot shows a directory listing for 'sha1sum \*.pdf' with two files having identical hashes: '38762cf7f55934b34d179ae6a4c80cadccbb7f0a 1.pdf' and '38762cf7f55934b34d179ae6a4c80cadccbb7f0a 2.pdf'. A subdirectory '/tmp/sha1' is also shown, containing 'sha256sum \*.pdf' with two files having identical hashes: '2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0 1.pdf' and 'd4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff 2.pdf'. A progress bar at the bottom right of the terminal shows '0.64G' and '8-11h'.

# الگوریتم‌های شکسته شده

## ▪ الگوریتم MD2

\* این الگوریتم در سال 1989 توسط آقای Ronald Linn Rivest معرفی شد. این الگوریتم برای کامپیوترهای ۸-بیتی بهینه‌سازی شده است

\* این الگوریتم نیز تا سال ۲۰۰۳ تضعیف شده بود و در نهایت، در سال ۲۰۰۴، ۲۰۰۸ و ۲۰۰۹ مورد حمله قرار گرفت. دو حمله اول از نوع پیش‌تصویر و حمله آخر از نوع تصادم می‌باشد

\* این الگوریتم از سال ۲۰۰۴ به طور کامل شکسته شده و در سال ۲۰۰۹ هم در OpenSSL، GnuTLS و NSS غیرفعال گردید ولی هنوز در زیرساخت‌های کلید عمومی که برای تولید گواهینامه‌ها به کار می‌رود، مورد استفاده قرار می‌گیرد

# الگوریتم‌های شکسته شده

## ▪ الگوریتم MD4

\* این الگوریتم نیز توسط آقای Ronald Linn Rivest در سال ۱۹۹۰ معرفی شد. طول رشته درهم‌شده،

۱۲۸ بیت می‌باشد

\* تنها بعد از یک سال از معرفی، ضعف‌های الگوریتم در سال ۱۹۹۱ منتشر شد. همچنین در سال‌های ۱۹۹۵

و ۲۰۰۴ مورد حمله تصادم و در سال‌های ۲۰۰۸ و ۲۰۱۰ مورد حمله پیش‌تصویر قرار گرفت و مقاومت آن

شکسته شد

### MD4 collision example [\[ edit \]](#)

Let:

```
k1 = 839c7a4d7a92cb5678a5d5b9eea5a7573c8a74deb366c3dc20a083b69f5d2a3bb3719dc69891e9f95e809fd7e8b23ba6318ed45e51fe39708bf9427e9c3e8b9
k2 = 839c7a4d7a92cb678a5d529eea5a7573c8a74deb366c3dc20a083b69f5d2a3bb3719dc69891e9f95e809fd7e8b23ba6318edc45e51fe39708bf9427e9c3e8b9
```

$k1 \neq k2$ , but  $MD4(k1) = MD4(k2) = 4d7e6a1defa93d2dde05b45d864c429b$

# الگوریتم‌های شکسته شده

## ▪ الگوریتم MD5

\* این الگوریتم در سال ۱۹۹۲ منتشر و معرفی شد. در ابتدا جهت رمزنگاری مورد استفاده قرار می‌گرفت اما به دلیل تهدیدات امنیتی زیادی که مورد توجه آن قرار گرفت، به کلی کنار گذاشته شد اما هم‌اکنون هم برای بررسی یکپارچگی فایل‌ها (checksum) مورد استفاده قرار می‌گیرد

\* برای اولین بار در سال ۲۰۰۴، تصادم برای آن ارائه شد. این الگوریتم هر چند که در مقاله‌های دیگری نیز مورد حمله‌های تصادم و پیش‌تصویر قرار گرفته است ولی هنوز در برخی از سیستم‌ها، به عنوان درهم‌کننده

رمز عبور کاربران استفاده می‌شود

```
Sequence #1
d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
2f ca b5 87 12 46 7e ab 40 04 58 3e b8 fb 7f 89
55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 71 41 5a
08 51 25 e8 f7 cd c9 9f d9 1d bd f2 80 37 3c 5b
d8 82 3e 31 56 34 8f 5b ae 6d ac d4 36 c9 19 c6
dd 53 e2 b4 87 da 03 fd 02 39 63 06 d2 48 cd a0
e9 9f 33 42 0f 57 7e e8 ce 54 b6 70 80 a8 0d 1e
c6 98 21 bc b6 a8 83 93 96 f9 65 2b 6f f7 2a 70

Sequence #2
d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
2f ca b5 07 12 46 7e ab 40 04 58 3e b8 fb 7f 89
55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 f1 41 5a
08 51 25 e8 f7 cd c9 9f d9 1d bd 72 80 37 3c 5b
d8 82 3e 31 56 34 8f 5b ae 6d ac d4 36 c9 19 c6
dd 53 e2 34 87 da 03 fd 02 39 63 06 d2 48 cd a0
e9 9f 33 42 0f 57 7e e8 ce 54 b6 70 80 28 0d 1e
c6 98 21 bc b6 a8 83 93 96 f9 65 ab 6f f7 2a 70

Both produce MD5 digest 79054025255fb1a26e4bc422aef54eb4
```

# الگوریتم‌های شکسته شده

## ▪ الگوریتم RIPEMD

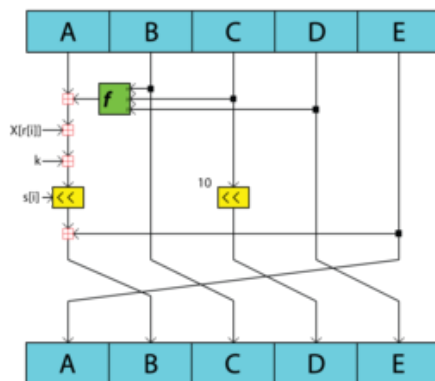
\* نمونه ابتدایی پیاده‌سازی این الگوریتم در سال ۱۹۹۲ و نمونه‌های دیگر آن در سال ۱۹۹۶ معرفی شدند. در

حال حاضر، نمونه RIPEMD-160 متداول بوده و به صورت عمومی مورد استفاده قرار می‌گیرد

\* در سال ۲۰۰۴، نمونه تصادم برای آن معرفی شد اما قابل اعمال برای نمونه دیگر آن (RIPEMD-160)

نبود

\* از نمونه توسعه یافته این الگوریتم برای تولید آدرس‌های بیت‌کوین نیز استفاده می‌شود



یک زیربلوک از تابع فشرده‌سازی الگوریتم  
درهم‌سازی RIPEMD-160

# الگوریتم‌های شکسته شده

## ▪ الگوریتم HAVAL

- \* این الگوریتم در سال ۱۹۹۲ معرفی شد. برخلاف الگوریتم MD5، این الگوریتم می‌توان رشته‌های درهم شده با طول متفاوتی را تولید کند. همچنین امکان مشخص کردن تعداد round ها نیز توسط کاربر وجود دارد
- \* خروجی‌های این الگوریتم به صورت اعداد هگزادسیمال ۳۲، ۴۰، ۴۸، ۵۶ و ۶۰ بیتی نشان داده می‌شوند
- \* این الگوریتم نیز مشابه MD5 و RIPEMD در سال ۲۰۰۴ و در همان مقاله، شکسته شد و برای آن یک تصادم ارائه شد

# الگوریتم‌های شکسته شده

## ▪ الگوریتم Snefru

\* این الگوریتم در سال ۱۹۹۰ توسط Ralph Merkle اختراع شد. خروجی‌های این الگوریتم به صورت ۱۲۸ بیتی و ۲۵۶ بیتی هستند

\* در سال ۲۰۰۸، با بهره‌گیری از تحلیل رمزی افتراقی، تصادم برای آن کشف شد

\* طراحی الگوریتم بعد از این اتفاق دستخوش تغییر شد و تعداد پیمایش‌های اصلی الگوریتم از ۲ به ۸ واحد افزایش یافت. هر چند که تحلیل رمزی افتراقی می‌تواند نسخه بهبود یافته را نیز شکست دهد اما تعداد عملیات‌های آن به قدری زیاد است که در عمل امکان‌پذیر نیست

# الگوریتم‌های شکسته شده

▪ طول عمر الگوریتم‌های بررسی شده

**Lifetimes of popular cryptographic hashes (the rainbow chart)**

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Snefru																												
MD2 (128-bit)[1]																												
MD4																												
MD5																												
RIPEMD																												
HAVAL-128[1]																												
SHA-0																												
SHA-1																												
RIPEMD-160																												
SHA-2 family																												
SHA-3 (Keccak)																												

**Key** Didn't exist/not public Under peer review Considered strong Minor weakness Weakened Broken Collision found



## با تشکر از توجه شما