# 现代密码学

# Modern Cryptography

张方国

中山大学计算机学院

Office: Room 305, IM School Building

E-mail: isszhfg@mail.sysu.edu.cn

HomePage: https://cse.sysu.edu.cn/content/2460

# 第十二讲 Hash函数（二）

- 迭代Hash函数
- MD结构
- 安全Hash函数
- SHA-3简介

# 迭代Hash函数

我们将使用压缩函数来构造hash函数，这样的hash函数被称为迭代Hash函数。

设$compress: \{0,1\}^{m+t} \to \{0,1\}^m$是一个压缩函数。我们基于压缩函数compress构造一个迭代Hash函数

$$h: \bigcup_{i=m+t+1}^{\infty} \{0,1\}^i \to \{0,1\}^l$$

1：预处理

2：处理

3：输出变换

# 迭代Hash函数

**预处理**

给定一个输入比特串$x$，其中$|x| \geq m+t+1$，用一个公开的算法构造一个串$y$，使得$|y| \equiv 0(mod\ t)$。记为

$$y = y_1||y_2||\cdots||y_r$$

其中对$1 \leq i \leq r$，有$|y_i| = t$。

预处理的通常用以下方式构造串$y$：

$$y = x||pad(x)$$

其中$pad(x)$是由填充函数对$x$作用后得到的。一个典型的填充函数是填入$|x|$的值，并填充一些额外的比特，使得所得到的比特串$y$变成$t$倍长。而且$x \to y$必须是1对1的，否则将不是碰撞稳固的。

# 迭代Hash函数

**处理**

设$IV$是一个长度为$m$的公开的初始值比特串。则计算：

$$z_0 \leftarrow IV$$

$$z_1 \leftarrow compress(z_0||y_1)$$

$$z_2 \leftarrow compress(z_1||y_2)$$

$$\vdots$$

$$z_r \leftarrow compress(z_{r-1}||y_r)$$

**输出变换**

设$g:\{0,1\}^m \rightarrow \{0,1\}^l$是一个公开函数。定义$h(x)=g(z_r)$。函数$g$是可选的。

# 一种特定的Hash函数结构：MD

- Merkle-Damgard结构

设$compress: \{0,1\}^{m+t} \to \{0,1\}^m$是一个碰撞稳固压缩函数。我们利用compress构造一个碰撞稳固的Hash函数

$$h: \bigcup_{i=m+t+1}^{\infty} \{0,1\}^i \to \{0,1\}^m$$

假定$t \geq 2$，$|x| = n \geq m+t+1$。把$x$表示成串联的形式

$$x = x_1 || x_2 || \cdots || x_k$$

其中$|x_1| = \cdots = |x_{k-1}| = t-1$，$|x_k| = t-1-d$（$0 \leq d \leq t-2$）。因此$k = \lceil \frac{n}{t-1} \rceil$

# 一种特定的Hash函数结构：MD

**算法4.6** Merkle-Damgard(x)

external compress

$n \leftarrow |x|, k \leftarrow \lceil n/(t-1) \rceil, d \leftarrow n - k(t-1)$

for $i \leftarrow 1$ to $k-1$

do $y_i \leftarrow x_i$

$y_k \leftarrow x_k || 0^d$

$y_{k+1} \leftarrow d$的二进制表示

$z_1 \leftarrow 0^{m+1} || y_1$

$g_1 \leftarrow compress(z_1)$

for $i \leftarrow 1$ to $k$

$z_{i+1} \leftarrow g_i || 1 || y_{i+1}$ 和$g_{i+1} \leftarrow compress(z_{i+1})$

$h(x) \leftarrow g_{k+1}$

return$(h(x))$

# 一种特定的Hash函数结构：MD

**定理4.6** 假定$compress: \{0,1\}^{m+t} \to \{0,1\}^m$是一个碰撞稳固压缩函数，其中$t \geq 2$。则按照算法4.6构造的函数

$$h: \bigcup_{i=m+t+1}^{\infty} \{0,1\}^i \to \{0,1\}^m$$

是一个碰撞稳固的Hash函数。

# 一种特定的Hash函数结构：MD

如果$t=1$。我们定义一个编码函数，即$f(0)=0, f(1)=01$。该编码满足两个性质：1，单射；2，没有任何编码是其他码字的后缀。

**算法4.7** Merkle-Damgard2(x)

external compress

$n \leftarrow |x|, y \leftarrow 11||f(x_1)||f(x_2)||\cdots||f(x_n)$

令$y=y_1||y_2||\cdots||y_k$，其中$y_i \in \{0,1\}$，$1 \le i \le k$

$g_1 \leftarrow compress(0^m||y_1)$

for $i \leftarrow 1$ to $k-1$

$$g_{i+1} \leftarrow compress(g_i||y_{i+1})$$

return($g_k$)

**定理4.7** 假定$compress: \{0,1\}^{m+1} \to \{0,1\}^m$是一个碰撞稳固压缩函数。则按照算法4.7构造的函数

$$h: \bigcup_{i=m+2}^{\infty} \{0,1\}^i \to \{0,1\}^m$$

是一个碰撞稳固的Hash函数。

# 一种特定的Hash函数结构：MD

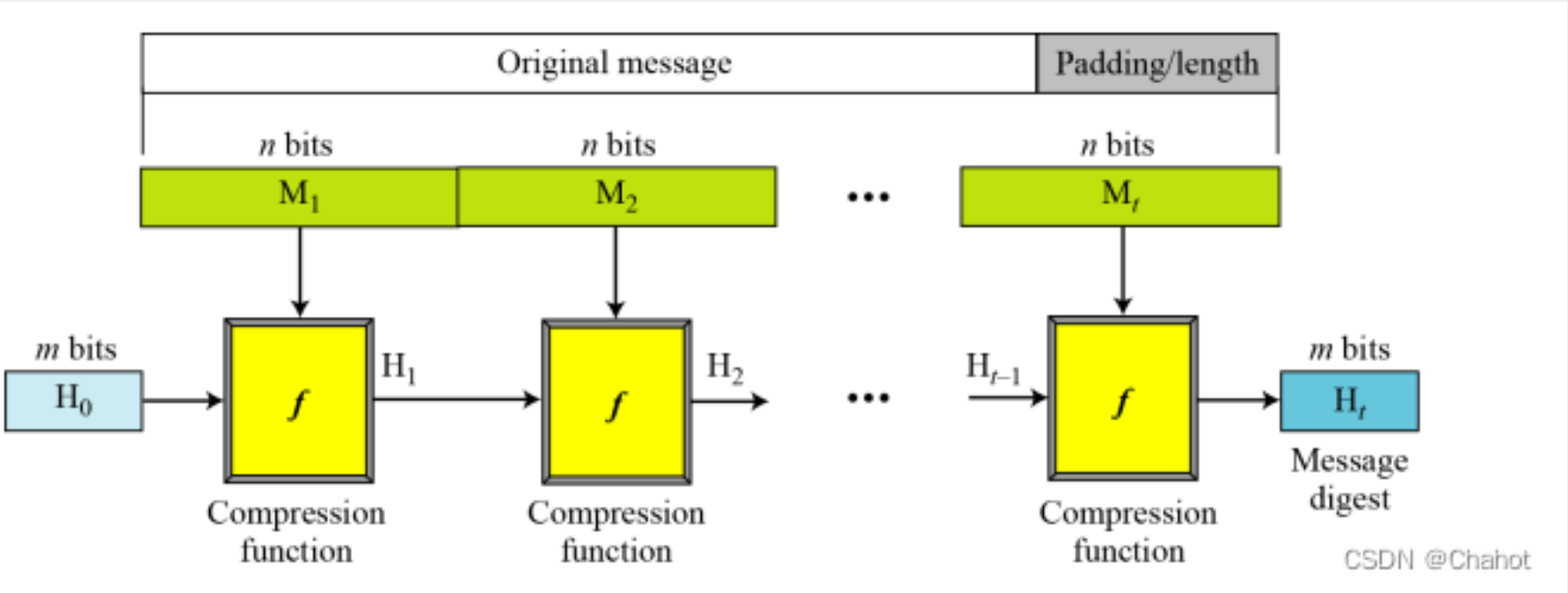
由定理4.6和4.7我们得到如下的结论：**定理4.8** 假定$compress: \{0,1\}^{m+t} \to \{0,1\}^m$是一个碰撞稳固压缩函数，其中$t \geq 1$。则存在一个碰撞稳固的Hash函数

$$h: \bigcup_{i=m+2}^{\infty} \{0,1\}^i \to \{0,1\}^m$$

。

在求$h$的值的时候，Compress最多被计算的次数为$1+\lceil \frac{n}{t-1} \rceil (t \geq 2)$或者$2n+2(t=1)$。

# MD


Original message
Padding/length
n bits
n bits
n bits
M1
M2
Mt
m bits
H0
f
H1
f
H2
Ht−1
f
m bits
Ht
Message digest
Compression function
Compression function
Compression function
CSDN @Chahot

# 安全哈希算法：SHA

本节主要介绍SHA-1（安全hash函数），这是一个具有160比特消息摘要的迭代hash函数。SHA-1所用到的操作如下：

| | |
|---|---|
| $X \wedge Y$ | $X$和$Y$的逻辑和 |
| $X \vee Y$ | $X$和$Y$的逻辑或 |
| $X \oplus Y$ | $X$和$Y$的逻辑异或 |
| $\neg X$ | $X$的逻辑补 |
| $X + Y$ | 模$2^{32}$整数加 |
| $ROTL^s(X)$ | $X$循环左移$s$个位置（$0 \le s \le 31$） |

# 安全哈希算法：SHA

SHA-1的填充方案。

**算法4.8** SHA-1-PAD(x)

注释$|x| \leq 2^{64} - 1$

$d \leftarrow (447 - |x|) mod\ 512$

$l \leftarrow |x|$的二进制表示，其中$|l| = 64$

$y \leftarrow x||1||0^d||l$

在$y$的构造中，添加了一个单独的1给$x$，然后串联足够的0使得长度为模512同余448，最后串联包括$x$的长度的二进制表示的64比特。所得的串$y$的长度能被512整除。所以，$y$可以写成由每个分组为512比特，共$n$个分组组成的串联：

$$y = M_1||M_2||\cdots||M_n$$

# 安全哈希算法：SHA

按照如下的方式定义$\mathbf{f}_0,\ldots,\mathbf{f}_{79}$：

$$f_t(B,C,D)=\begin{cases}(B\wedge C)\vee((\neg B)\wedge D) & 0\le t\le 19\\ B\oplus C\oplus D & 20\le t\le 39\\ (B\wedge C)\vee(B\wedge D)\vee(C\wedge D) & 40\le t\le 59\\ B\oplus C\oplus D & 60\le t\le 79\end{cases}$$

以及定义常数$K_0,\ldots,K_{79}$：

$$K_t=\begin{cases}5A827999 & 0\le t\le 19\\ 6ED9EBA1 & 20\le t\le 39\\ 8F1BBCDC & 40\le t\le 59\\ CA62C1D6 & 60\le t\le 79\end{cases}$$

# 安全哈希算法：SHA

密码体制**4.1** SHA-1(x)

external SHA-1-PAD

global $K_0, \ldots, K_{79}$

$y \leftarrow$ SHA-1-PAD(x)，令$y = M_1||M_2||\cdots||M_n$

$H_0 \leftarrow 67452301$, $H_1 \leftarrow EFCDAB89$, $H_2 \leftarrow 98BADCFE$,

$H_3 \leftarrow 10325476$, $H_4 \leftarrow C3D2E1F0$

for $i \leftarrow 1$ to $n$

$$do \begin{cases} Let\ M_i = W_0||W_1||\cdots||W_{15} \\ for\ t \leftarrow 16\ to\ 79 \\ \qquad do\ W_i \leftarrow ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \\ A \leftarrow H_0,\ B \leftarrow H_1,\ C \leftarrow H_2,\ D \leftarrow H_3,\ E \leftarrow H_4 \\ (unfinshed) \end{cases}$$

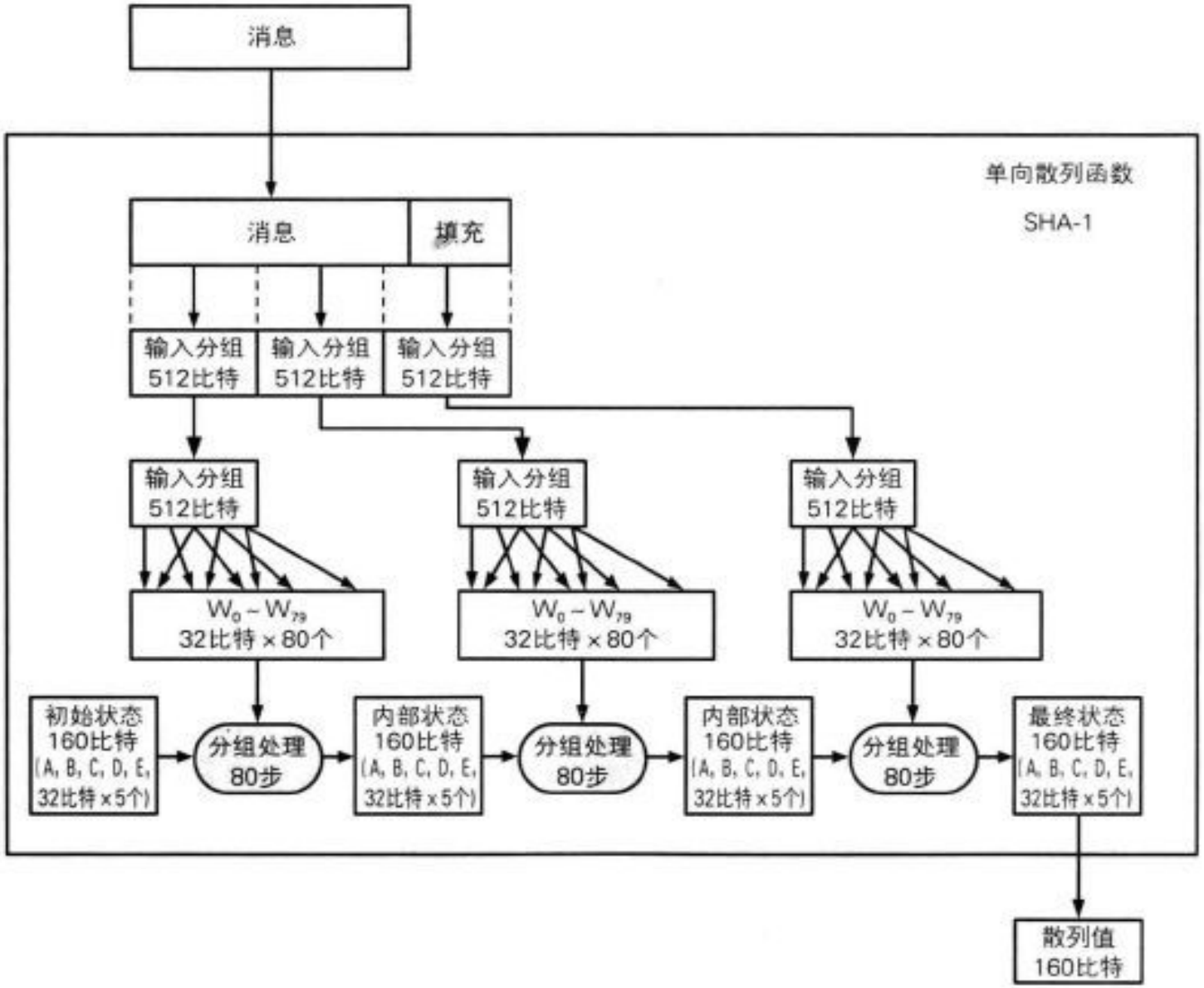
# 安全哈希算法：SHA

密码体制**4.1** SHA-1(x)

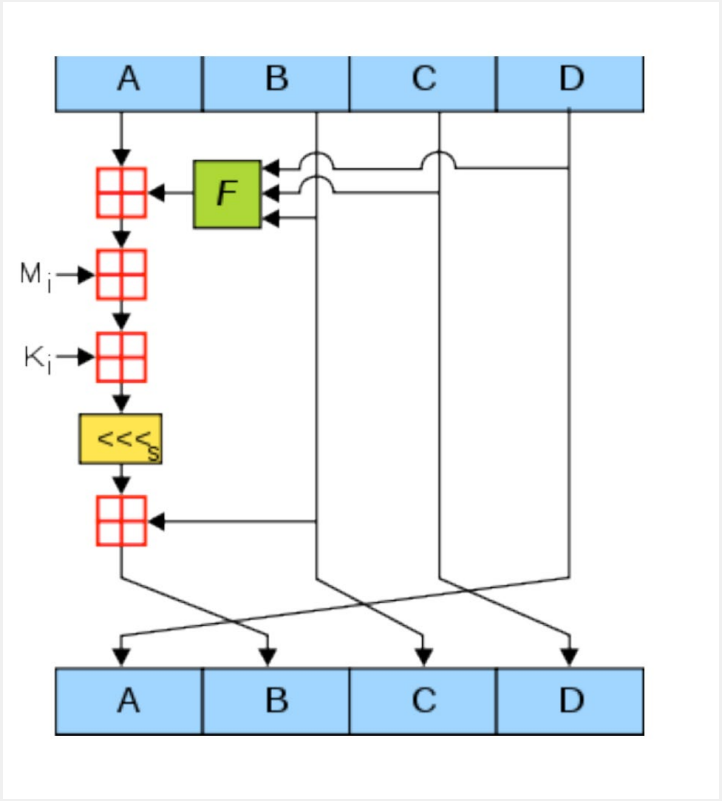
$$
do\begin{cases}
for\ t \leftarrow 0\ to\ 79 \\
\quad (1)\ temp \leftarrow ROTL^5(A) + f_t(B,C,D) + E + W_t + K_t \\
\quad (2)\ E \leftarrow D \\
\quad (3)\ D \leftarrow C \\
\quad (4)\ C \leftarrow ROTL^{30}(B) \\
\quad (5)\ B \leftarrow A \\
\quad (6)\ A \leftarrow temp \\
H_0 \leftarrow H_0 + A, H_1 \leftarrow H_1 + B, H_2 \leftarrow H_2 + C \\
H_3 \leftarrow H_3 + D, H_4 \leftarrow H_4 + E
\end{cases}
$$

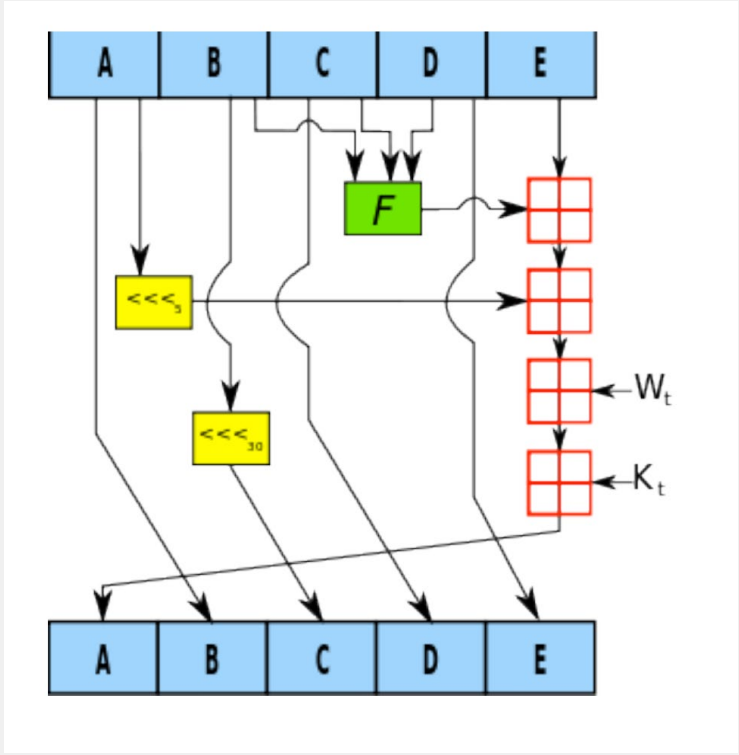
return$(H_0||H_1||H_2||H_3||H_4)$

# SHA-1总结


消息
单向散列函数
SHA-1
消息
填充
输入分组
512比特
输入分组
512比特
输入分组
512比特
输入分组
512比特
输入分组
512比特
输入分组
512比特
$W_0 - W_{79}$
32比特×80个
$W_0 - W_{79}$
32比特×80个
$W_0 - W_{79}$
32比特×80个
初始状态
160比特
(A, B, C, D, E,
32比特×5个)
分组处理
80步
内部状态
160比特
(A, B, C, D, E,
32比特×5个)
分组处理
80步
内部状态
160比特
(A, B, C, D, E,
32比特×5个)
分组处理
80步
最终状态
160比特
(A, B, C, D, E,
32比特×5个)
散列值
160比特

# MD5和SHA-1



# HASH灾难年！

## Crypto 2004

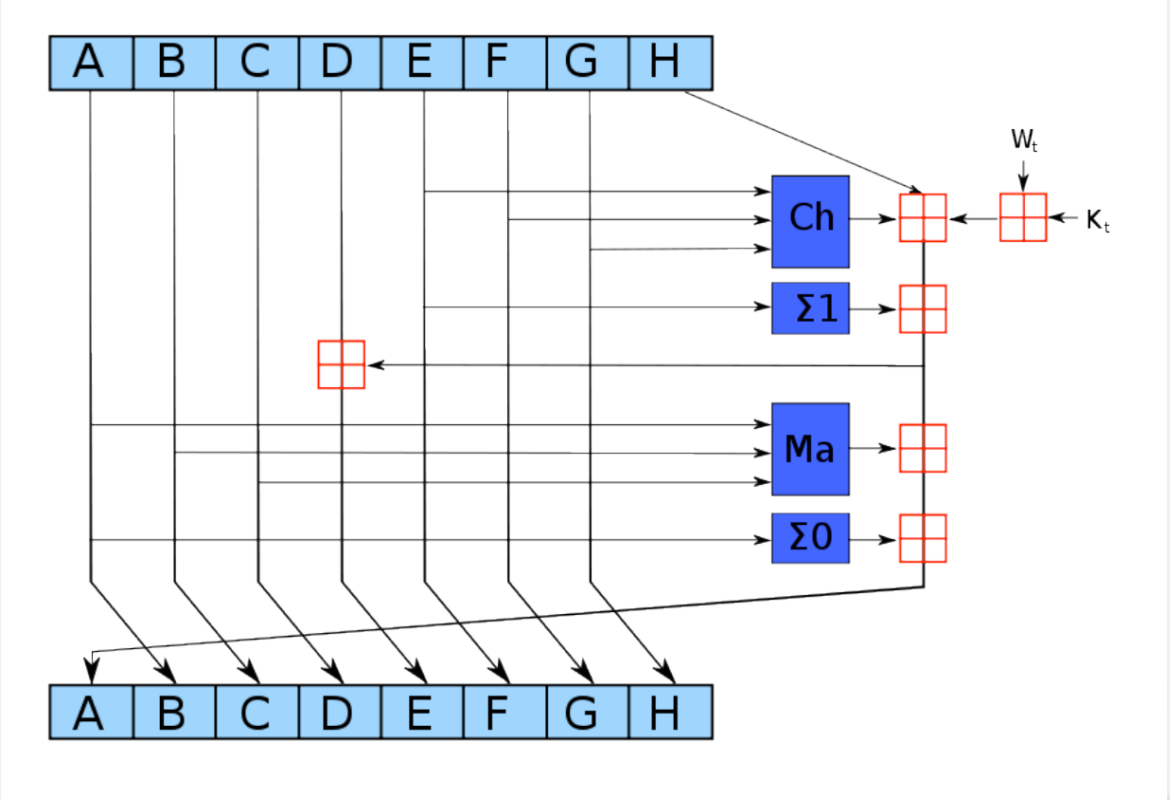
- Conference:
- Joux shows a surprising property in Merkle-Damgaard hashes
  - Multicollisions
  - Cascaded hashes don't help security much
- Biham/Chen attack SHA-0 (neutral bits)
- Rump Session:
- Joux shows attack on SHA-0
- Wang shows attacks on MD4, MD5, RIPEMD, some Haval variants, and SHA-0
  - Much better techniques used for these attacks

# SHA-1的碰撞

- 2005年二月，王小云、殷益群及于红波发表了对完整版SHA-1的攻击，只需少于2^58的计算复杂度，就能找到一组碰撞
- 2006年的CRYPTO会议上，Christian Rechberger和Christophe De Cannière宣布他们能在**容许攻击者决定部分原消息的条件之下**，找到SHA-1的一个碰撞。
- 2017年2月23日，Google公司公告宣称他们与CWI Amsterdam合作共同创建了两个有着相同的SHA-1值但内容不同的PDF文件，这代表SHA-1算法已被正式攻破。

# 过渡：SHA-2

- 2001年颁布
- SHA-256
- SHA-384
- SHA-512



# SHA-3

- 美国国家标准技术研究协会（NIST）于2007年宣布了Hash算法竞赛，寻找SHA-2的替代，最终将于2012年选出获胜算法并命名为**SHA-3**

http://csrc.nist.gov/groups/ST/hash/sha-3/index.html

- 截止2008年11月，一共有64个算法提交。
- 2008年12月其中51个算法被NIST接收作为第一轮的候选算法。
- 2009年7月，NIST公布了进入第二轮的14个算法。

| | |
|---|---|
| BLAKE | Jean-Philippe Aumasson |
| Blue Midnight Wish | Svein Johan Knapskog |
| CubeHash | Daniel J. Bernstein |
| ECHO | Henri Gilbert |
| Fugue | Charanjit S. Jutla |
| Grøstl | Lars R. Knudsen |
| Hamsi | Özgül Küçük |
| JH | Hongjun Wu |
| Keccak | The Keccak Team |
| Luffa | Dai Watanabe |
| Shabal | Jean-François Misarsky |
| SHAvite-3 | Orr Dunkelman |
| SIMD | Gaëtan Leurent |
| Skein | Bruce Schneier |

# 最后一轮五名候选者的名单是

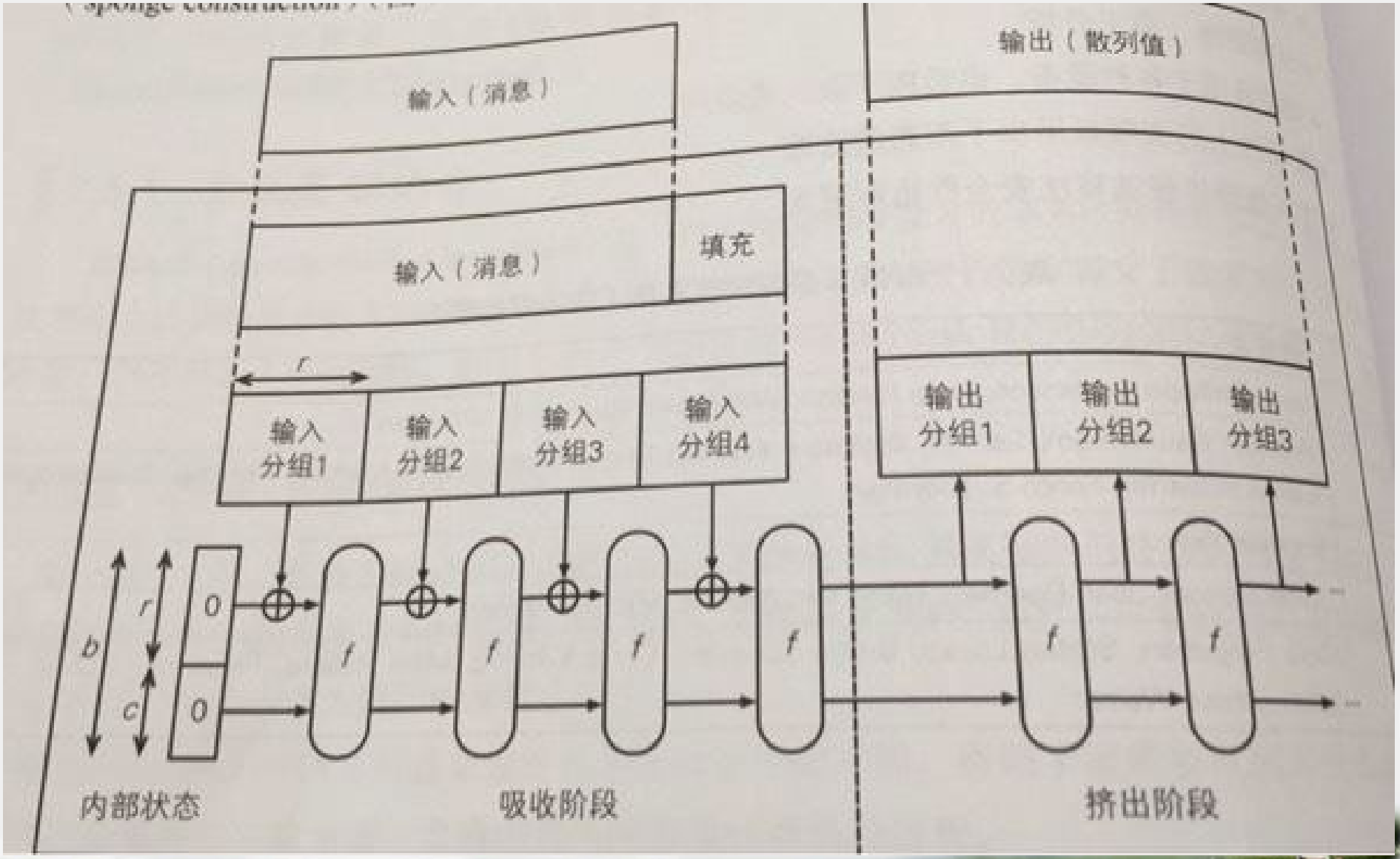
- 源自瑞士的BLAKE，
- Graz理工大学和丹麦理工大学合作的Grøstl
- 新加坡华裔信息安全专家伍宏军的JH，
- Joan Daemen为主要成员的Keccak，
- Bruce Schneier为负责人的Skein。
- http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/March2012/Program_SHA3_March2012.html

# SHA-3：Keccak

- 2012年10月2日，SHA-3获胜算法揭晓：Keccak
- Keccak is designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche 。
- NIST计算机安全专家Tim Polk说，Keccak的优势在于它与SHA-2设计上存在极大差别，适用于SHA-2的攻击方法将不能作用于Keccak。

Keccak团队: Michaël Peeters, Guido Bertoni, Joan Daemen,Ronny Van Keer and Gilles Van Assche(左至右)

# SHA-3: Keccak


输入（消息）
输入（消息）
填充
r
输入
分组1
输入
分组2
输入
分组3
输入
分组4
输出（散列值）
输出
分组1
输出
分组2
输出
分组3
b
r
c
0
0
f
f
f
f
f
f
f
内部状态
吸收阶段
挤出阶段

# SHA-3计算过程：填充，吸收，挤压