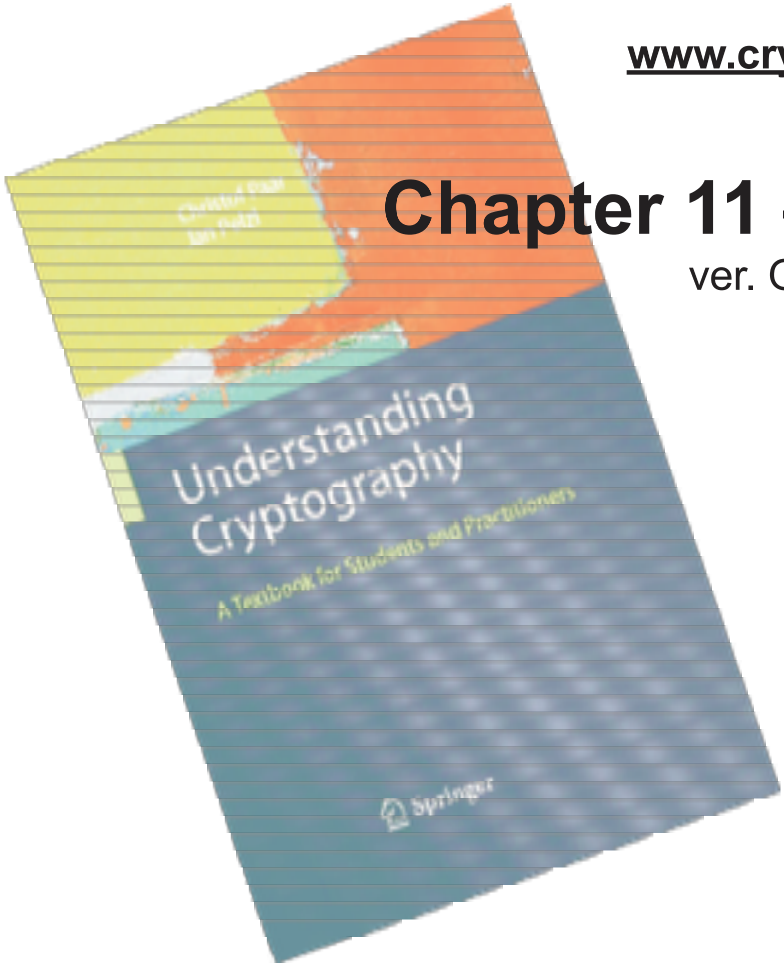
Understanding Cryptography – A Textbook for Students and Practitioners

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 11 – Hash Functions

ver. October 29, 2009

These slides were prepared by Stefan Heyse and Christof Paar and Jan Pelzl
And modified by Sam Bowne

Some legal stuff (sorry): Terms of Use

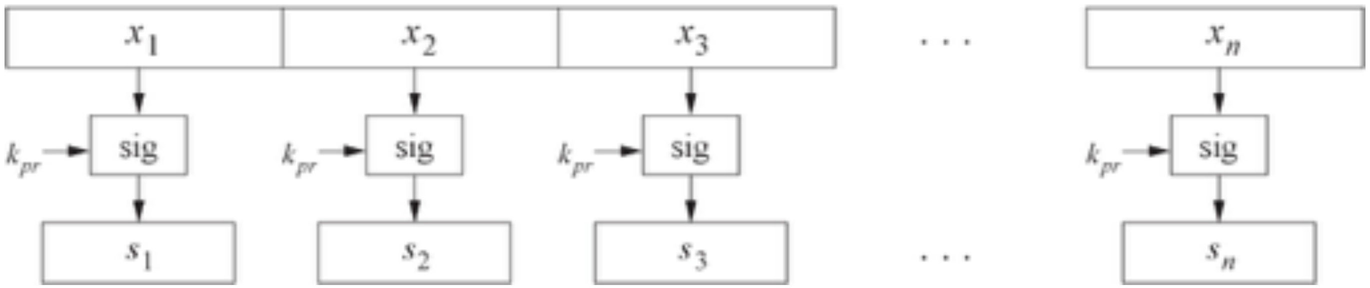
Contents of this Chapter

11.1 Motivation: Signing Long Messages

Motivation

Problem:

Naive signing of long messages generates a signature of same length.



- Three Problems
- Computational overhead
- Message overhead
- Security limitations
  - Attacker could re-order or re-use signed blocks

Solution:

Instead of signing the whole message, sign only a digest (=hash)

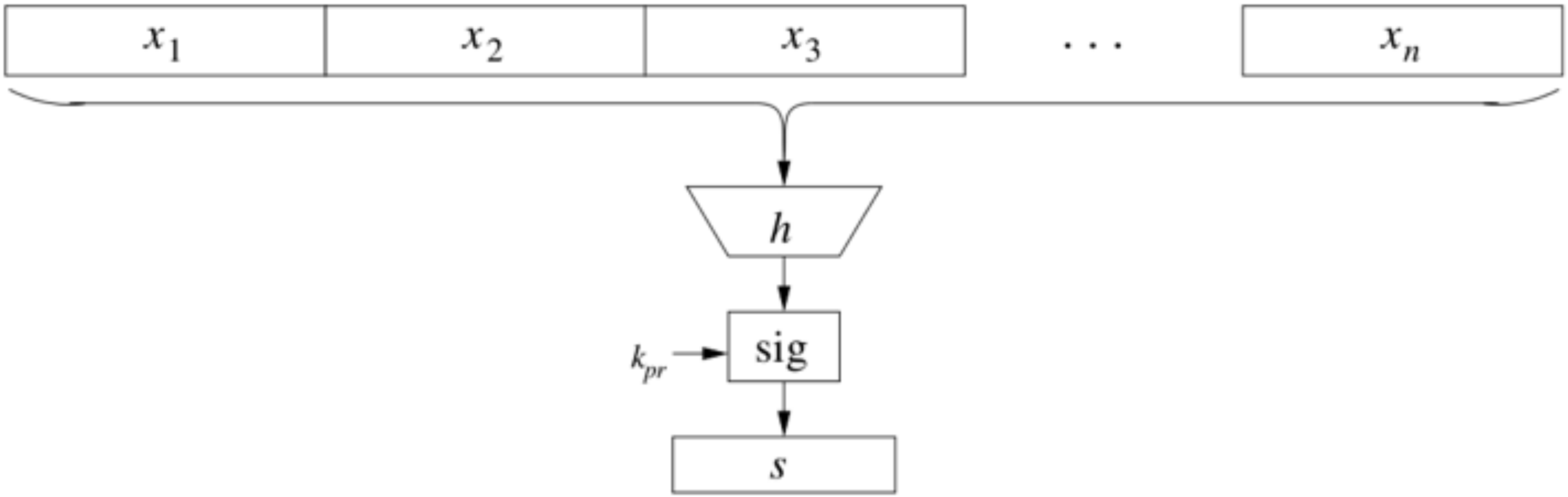
Also secure, but much faster

Needed:

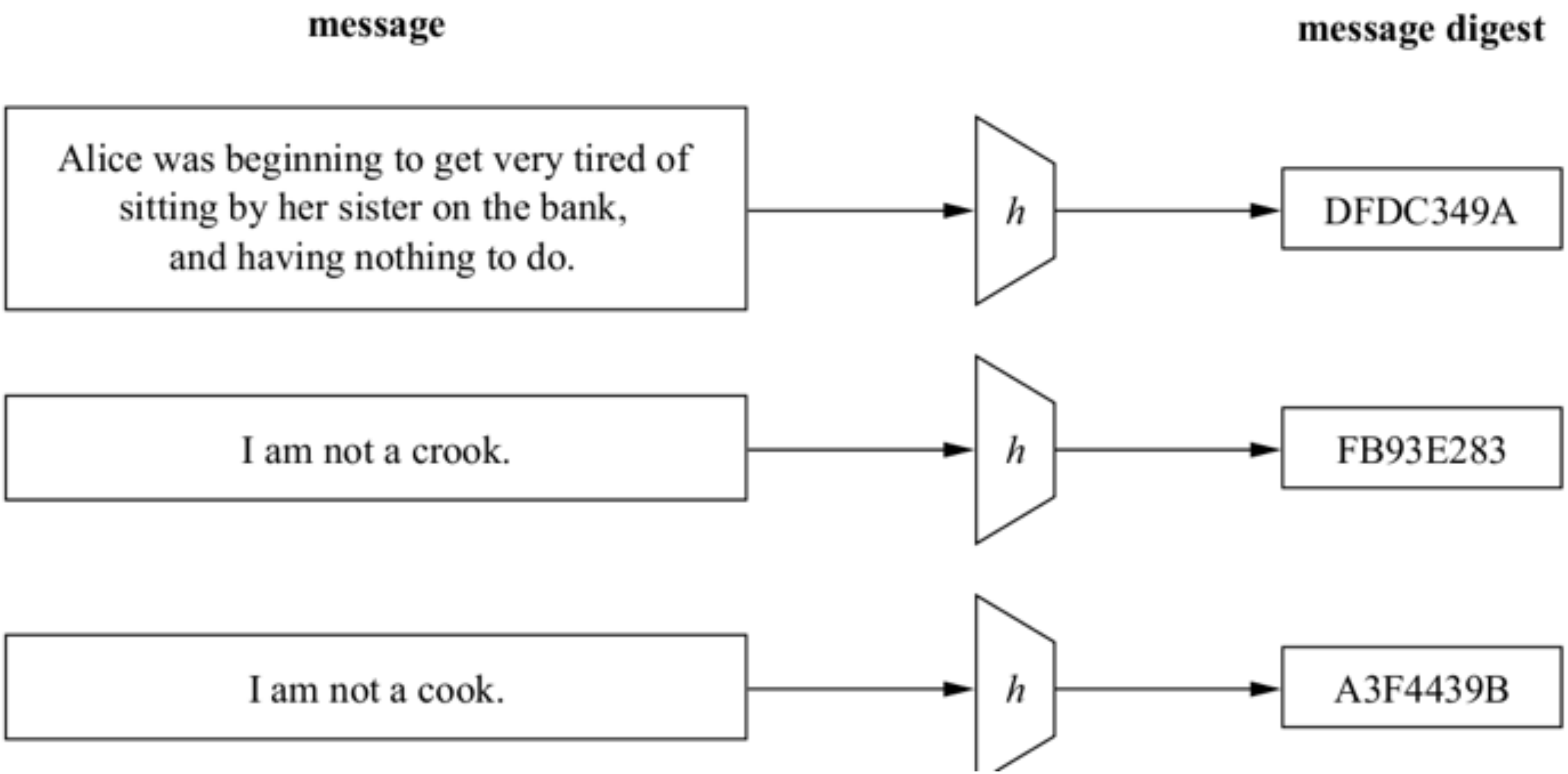
 Hash Functions

Solution

- Hash, then sign

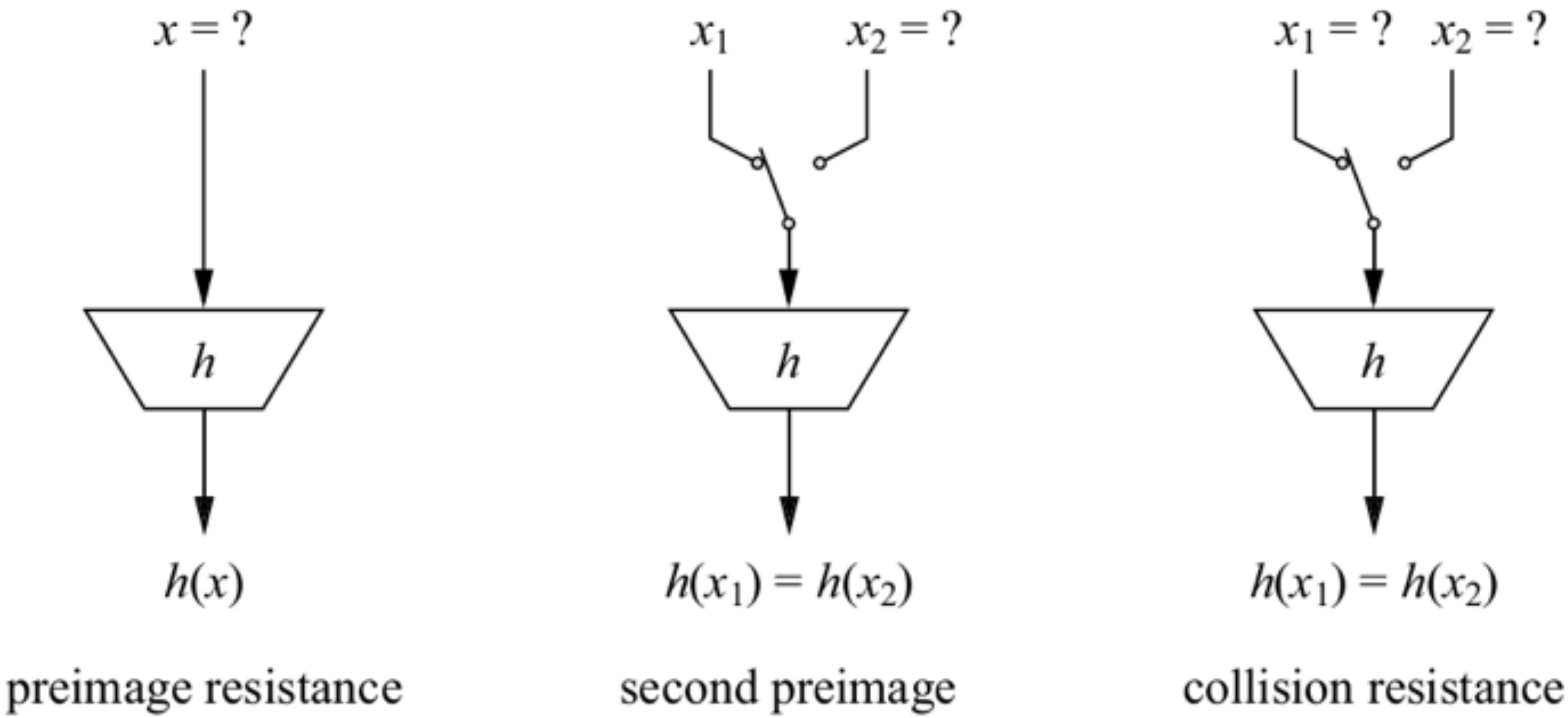


Principal input–output behavior of hash functions



11.2 Security Requirements of Hash Functions

The three security properties of hash functions



Hash Functions: Security Properties

- **Preimage resistance:** For a given output z, it is impossible to find any input x such that h(x) = z, i.e., h(x) is one-way (Also called **one-wayness**)
- **Second preimage resistance:** Given x_1, and thus $h(x_1)$, it is computationally infeasible to find any x_2 such that $h(x_1) = h(x_2)$ (Also called **weak collision resistance**)
- **Collision resistance:** It is computationally infeasible to find any pairs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$ (Also called **strong collision resistance**)

Hash Functions: Security

- Collison resistance causes most problems
- How hard is it to find a collision with a probability of 0.5 ?
- Related Problem: How many people are needed such that two of them have the same birthday with a probability of 0.5 ?
- No! Not 365/2=183
- 23 are enough ! This is called the birthday paradox (Search takes $\approx\sqrt{2n}$ steps)
- To deal with this paradox, hash functions need a output size of at least 160 bits

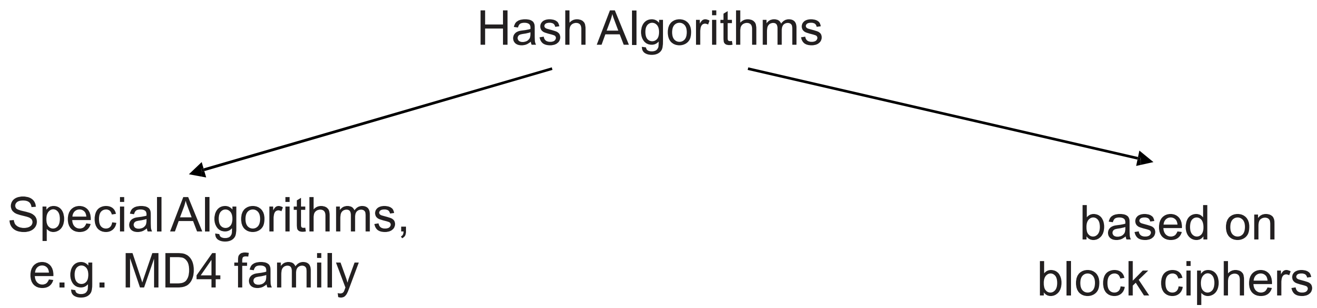
Hash Functions: Security

Table 11.1 Number of hash values needed for a collision for different hash function output lengths and for two different collision likelihoods

λ	Hash output length 128 bit	160 bit	256 bit	384 bit	512 bit
0.5	2^{65}	2^{81}	2^{129}	2^{193}	2^{257}
0.9	2^{67}	2^{82}	2^{130}	2^{194}	2^{258}

11.3 Overview of Hash Algorithms

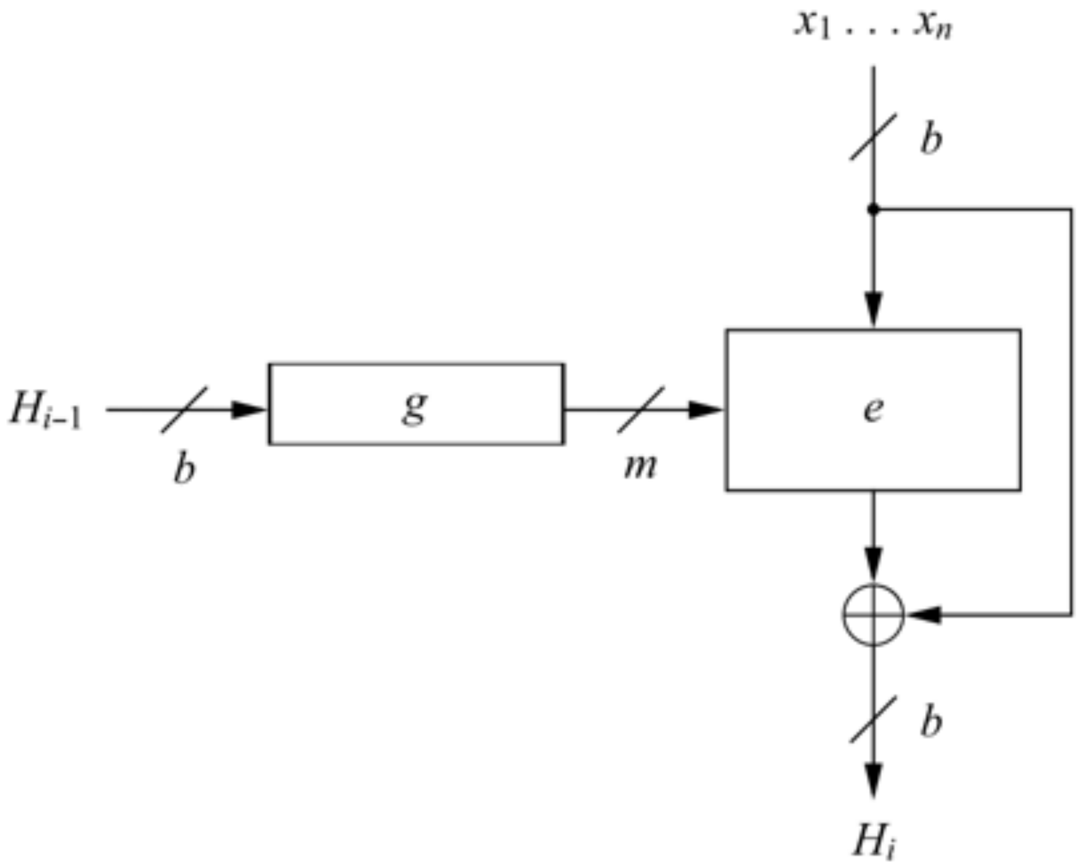
Hash Functions: Algorithms



- **MD4** family
- **SHA-1**: output - 160 Bit; input - 512 bit chunks of message *x;* operations - bitwise AND, OR, XOR, complement and cyclic shifts.
- **RIPE-MD 160:** output - 160 Bit; input - 512 bit chunks of message *x*; operations – like in SHA-1, but two in parallel and combinations of them after each round.

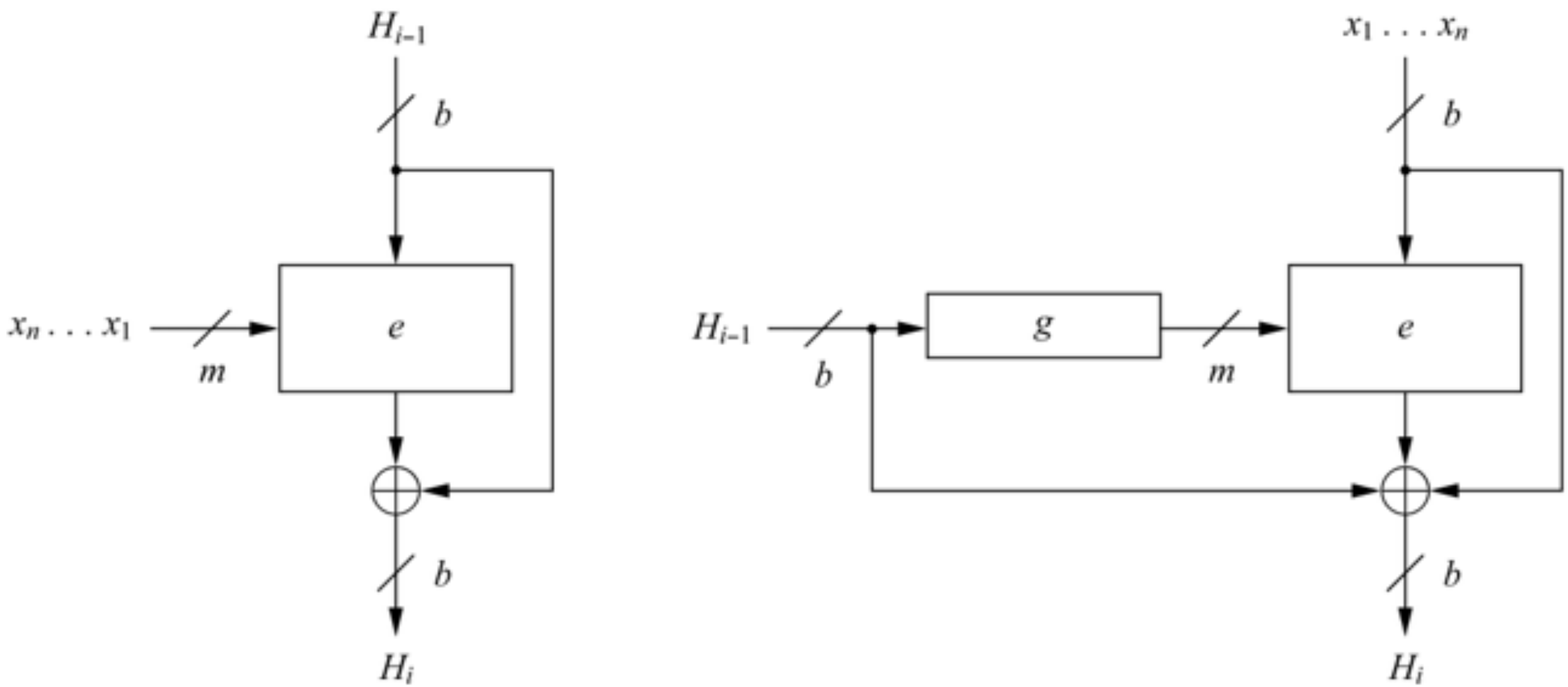
Hash Functions from Block Ciphers

- Matyas-Meyer-Oseas hash function
- Break original file into blocks
- Start with known public block H0, which may be all zeroes
- Encrypt a block using previous H as the key, then XOR with next block of file data to form the new H
- Repeat through whole file



$$H_i = e_{g(H_{i-1})}(x_i) \oplus x_i$$

Other Hash Functions from Block Ciphers



$$H_i = H_{i-1} \oplus e_{x_i}(H_{i-1}) \qquad \text{(Davies–Meyer)}$$

$$H_i = H_{i-1} \oplus x_i \oplus e_{g(H_{i-1})}(x_i) \qquad \text{(Miyaguchi–Preneel)}$$

11.4 The Secure Hash Algorithm SHA-1

SHA-1

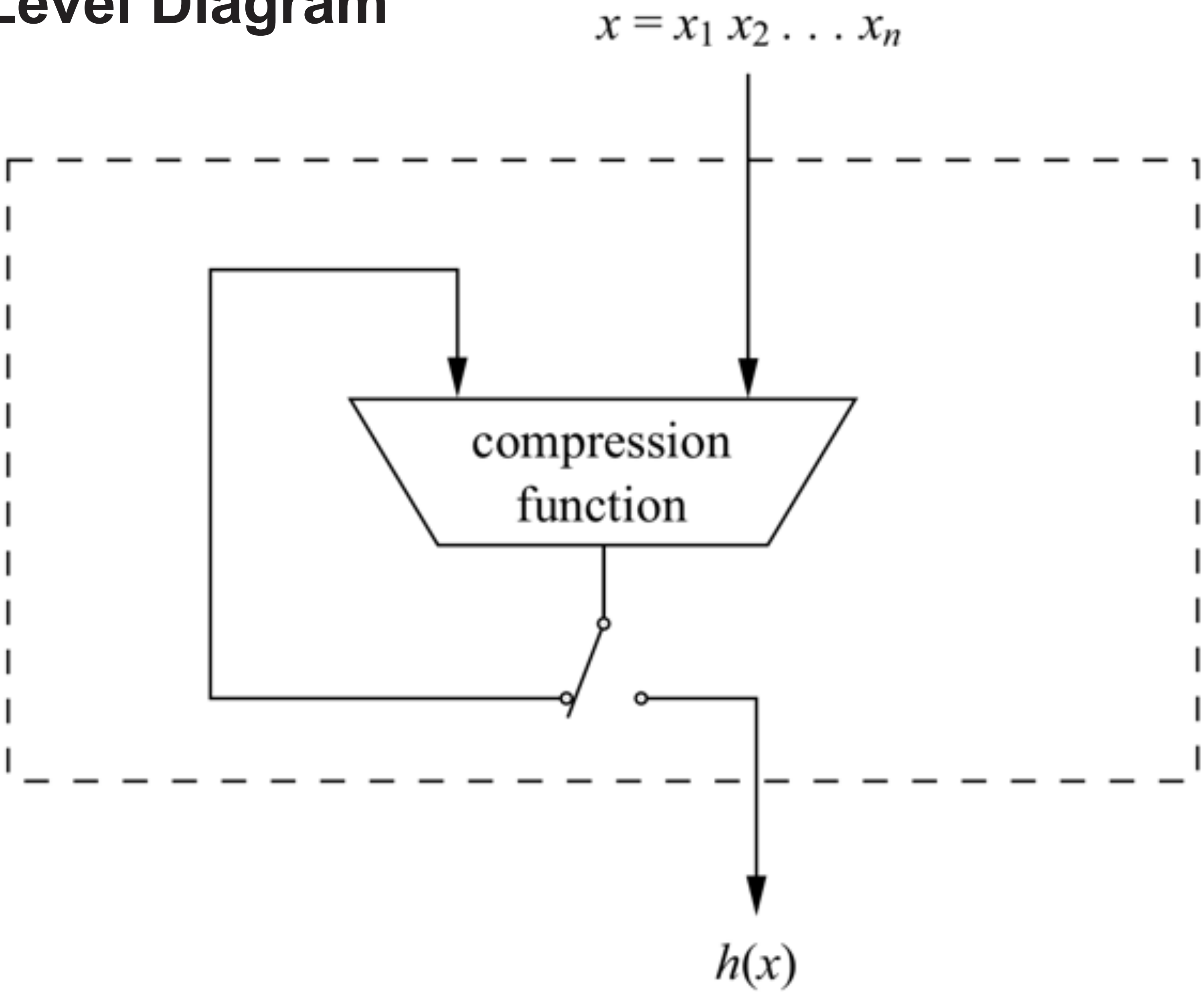
- Part of the MD-4 family.
- Based on a Merkle-Dåmgard construction
- Similar to a Feistel block cipher
- 160-bit output from a message of maximum length 2^{64} bit.
- Widely used (even tough some weaknesses are known)

How Big is 2**60 bits?

- 2**10 = 1024 = 1Kb
- 2**20 = 1Mb
- 2**30 = 1Gb
- 2**40 = 1 Tb
- 2**60 = 1 million Tb
- 2**63 = 1 million TB
- 2**54 = 2 million TB

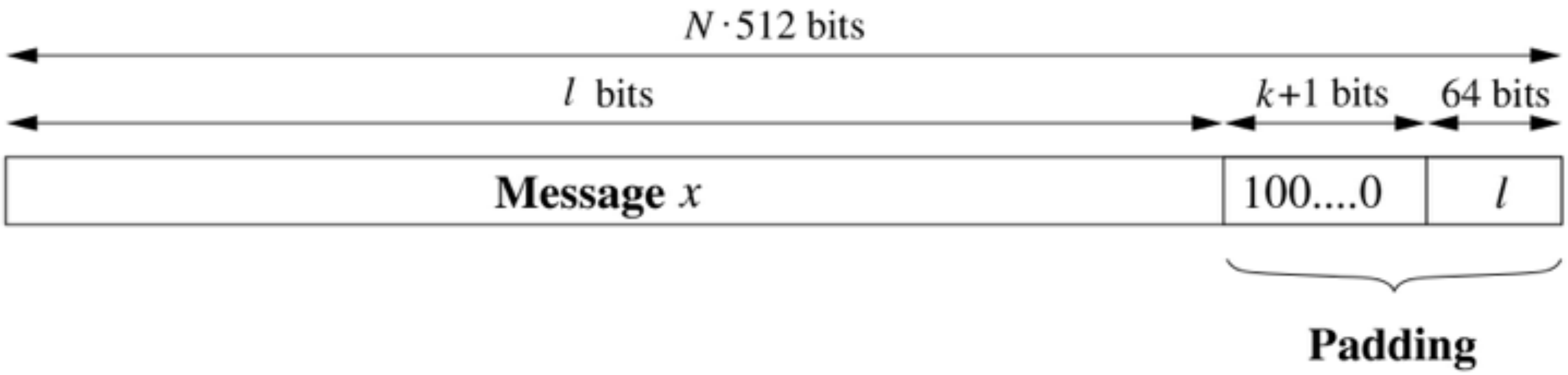
SHA-1 High Level Diagram

Compression Function consists of 80 rounds which are divided into four stages of 20 rounds each



SHA-1: Padding

- Message x has to be padded to fit a size of a multiple of 512 bits
- Add **k** zero bits
- **k** = 512 − 64 − 1 − l = 448 − (l + 1) mod 512



SHA-1: Padding Example

- Message is **abc**
  24 bits long

$$\underbrace{01100001}_{\text{a}} \quad \underbrace{01100010}_{\text{b}} \quad \underbrace{01100011}_{\text{c}}$$

k = 512 − 64 − 1 − 24 = 423

$$\underbrace{01100001}_{\text{a}} \quad \underbrace{01100010}_{\text{b}} \quad \underbrace{01100011}_{\text{c}} \quad 1 \quad \underbrace{00\ldots0}_{\text{423 zeros}} \quad \underbrace{00\ldots011000}_{\text{64 bits long}}$$

64 bits long
Value = 24 in binary

SHA-1: Hash Computation

- Each message block x_i is processed in four stages with 20 rounds each

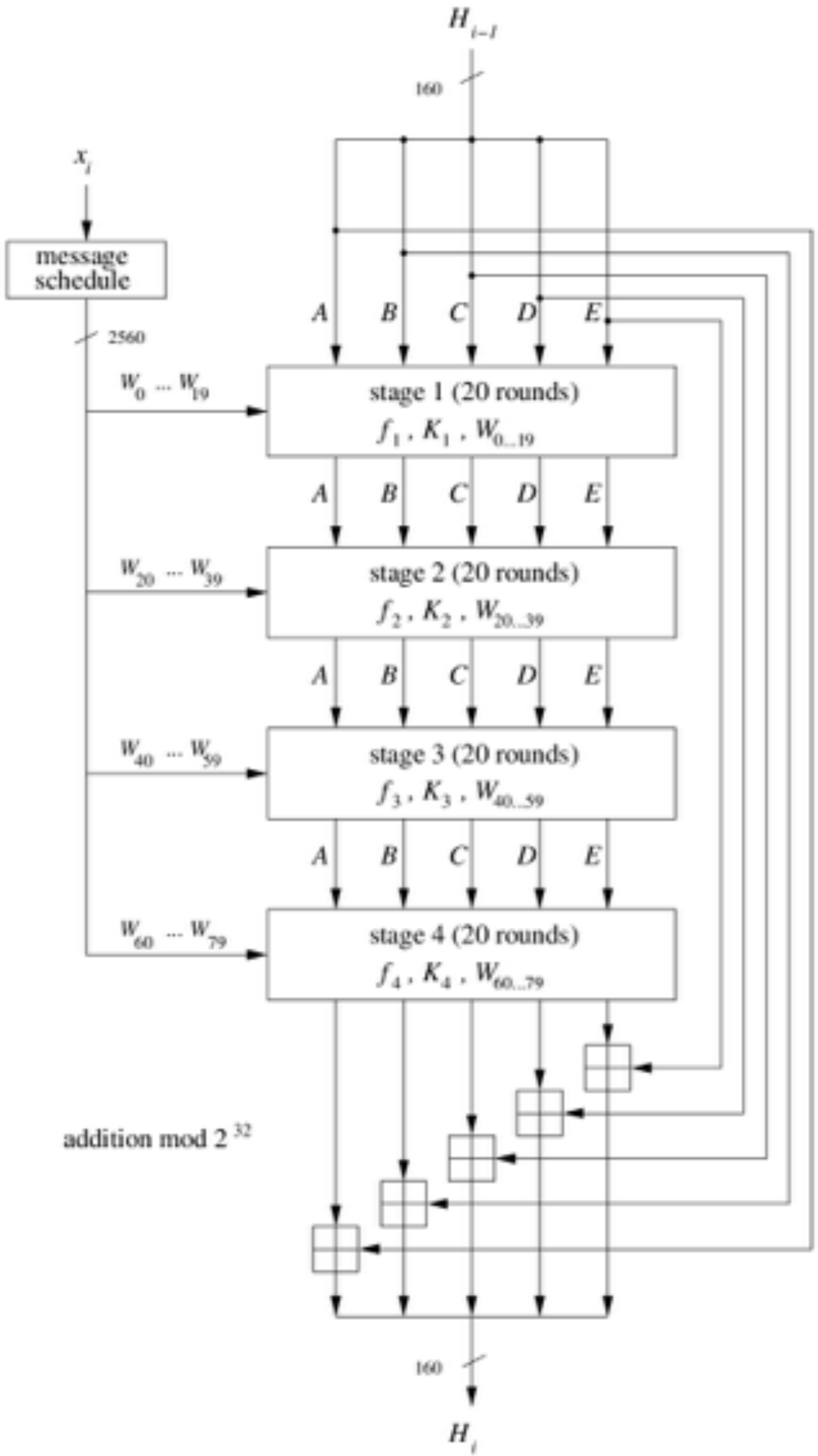
SHA-1 uses:

- A message schedule which computes a 32-bit word W0, W1, ..., W79 for each of the 80 rounds
- Five working registers of size of 32 bits A,B,C,D,E
- A hash value H_i consisting of five 32-bit words $H_i^{(0)}$, $H_i^{(1)}$, $H_i^{(2)}$, $H_i^{(3)}$, $H_i^{(4)}$
- In the beginning, the hash value holds the initial value H_0, which is replaced by a new hash value after the processing of each single message block.
- The final hash value H_n is equal to the output h(x) of SHA-1.

Initial value H_0 A 160-bit buffer is used to hold the initial hash value for the first iteration. The five 32-bit words are fixed and given in hexadecimal notation as:

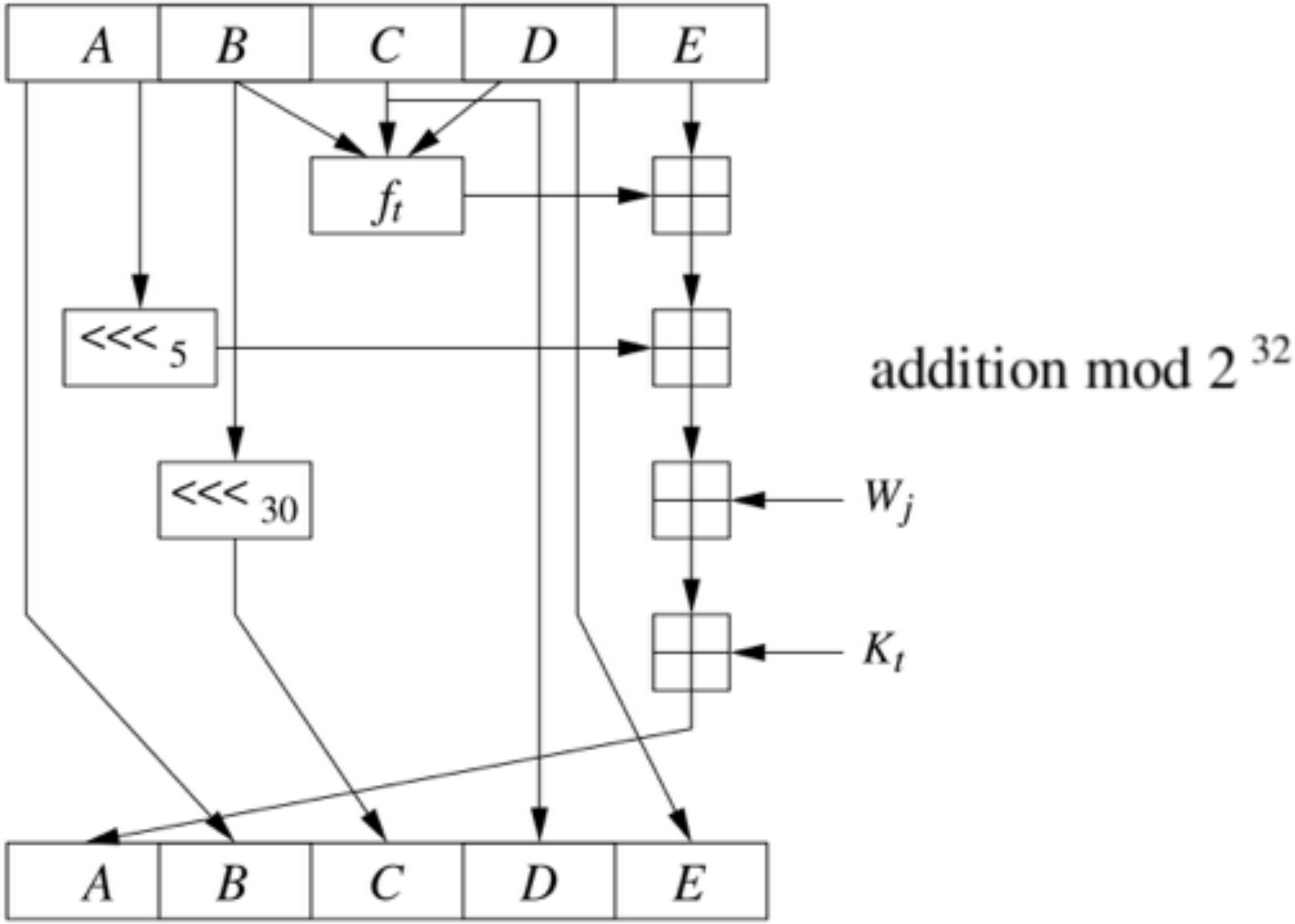
$$A = H_0^{(0)} = \texttt{67452301}$$
$$B = H_0^{(1)} = \texttt{EFCDAB89}$$
$$C = H_0^{(2)} = \texttt{98BADCFE}$$
$$D = H_0^{(3)} = \texttt{10325476}$$
$$E = H_0^{(4)} = \texttt{C3D2E1F0}.$$

SHA-1: All four stages



Chapter 11 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

SHA-1: Internals of a Round



Stage t	Round j	Constant K_t	Function f_t
1	00…19	5A827999	$(B \wedge C) \vee (\overline{B} \wedge D)$
2	20…39	6ED9EBA1	$B \oplus C \oplus D$
3	40…59	8F1BBCDC	$(B \oplus C) \vee (B \oplus D) \vee (C \oplus D)$
4	60…79	CA62C1D6	$B \oplus C \oplus D$

AND $\wedge$

OR $\vee$

NOT $\overline{}$

XOR $\oplus$

SHA-1 Collision Found

- Collision found on Feb. 23, 2017
  - Links Ch 12-2017-1, 2, and 3 in CNIT 123

```
Sams-MacBook-Pro-3:proj14 sambowne$ ls -l sha*
-rw-r--r--@ 1 sambowne  staff  422435 Feb 23  2017 shattered-1.pdf
-rw-r--r--@ 1 sambowne  staff  422435 Feb 23  2017 shattered-2.pdf
Sams-MacBook-Pro-3:proj14 sambowne$ shasum shattered-1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  shattered-1.pdf
Sams-MacBook-Pro-3:proj14 sambowne$ shasum shattered-2.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  shattered-2.pdf
Sams-MacBook-Pro-3:proj14 sambowne$ md5 shattered-1.pdf
MD5 (shattered-1.pdf) = ee4aa52b139d925f8d8884402b0a750c
Sams-MacBook-Pro-3:proj14 sambowne$ md5 shattered-2.pdf
MD5 (shattered-2.pdf) = 5bd9d8cabc46041579a311230539b8d1
```

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Announcing the first SHA1 collision

February 23, 2017

'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time

SHATTERED


Collision attack: same hashes
Good doc
Sha-1
3713..42
Bad doc
Sha-1
3713..42

Browsers Deprecated SHA-1

Microsoft, Google, and Mozilla will begin phasing out trust for SHA-1 certificates in 2016. With these dates approaching, it's time to move to SHA-2.

November 2014 –	*SHA-1 SSL Certificates expiring any time in 2017 will show a warning in Chrome.*
December 2014 –	*SHA-1 SSL Certificates expiring after June 1, 2016, will show a warning in Chrome.*
January 2015 –	*SHA-1 SSL Certificates expiring any time in 2016 will show a warning in Chrome.*
December 2015 –	*SHA-1 SSL Certificates issued after January 1, 2016, will show the "untrusted connection" error in Firefox.*
January 2016 –	*SHA-1 SSL Certificates issued after January 1, 2016, will show a certificate error in Chrome.* *Certificate criteria: signed with a SHA-1-base signature, issued after January 1, 2016, and chained to a public CA.*
January 1, 2017 –	*Microsoft, Google, and Mozilla will end trust for all SHA-1 SSL Certificates.* *Mozilla and Google say it is feasible to move this date up to July 1, 2016, in light of recent attacks on SHA-1.* *Microsoft says it is feasible to move this date up to as early as June 2016, in light of recent attacks on SHA-1.*

- Link Ch 12zr in CNIT 123

Lessons Learned: **Hash-Functions**

- Hash functions are keyless. The two most important applications of hash functions are their use in digital signatures and in message authentication codes such as HMAC.
- The three security requirements for hash functions are one-wayness, second preimage resistance and collision resistance.
- Hash functions should have at least 160-bit output length in order to withstand collision attacks; 256 bit or more is desirable for long-term security.
- MD5, which was widely used, is insecure. Serious security weaknesses have been found in SHA-1, and the hash function should be phased out. The SHA-2 algorithms all appear to be secure.
- The ongoing SHA-3 competition will result in new standardized hash functions in a few years.

Kahoot!