2: Iterated Cryptographic Hash Functions

- We want *hash function* $H : (\{0,1\}^n)^* \to \{0,1\}^n$ of *potentially infinite* input size
- Instead we have *compression function* $F : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ and define H via many calls to F
- One iterated construction (Merkle-Damgård):
  1. Fix an "initial value" $H_0 \in \{0,1\}^n$
  2. Write input M as a sequence of m-bit blocks: $M = (M_1, \ldots, M_L)$, $M_i \in \{0,1\}^m$.
  3. For i in $\{1, \ldots, L\}$ compute $H_i := F(H_{i-1}, M_i)$
  4. Output $H(M) = H_L$.
- Variant ("Merkle-Damgård strengthening"): change to

  2b. Write input M as a sequence of m-bit blocks: $M = (M_1, \ldots, M_L)$, $M_i \in \{0,1\}^m$, where M_L encodes the length L

  (Typical for first-generation hash functions)

The Benefits of Merkle-Damgård

Compression function secure $\Rightarrow$ hash function secure

Theorem 3

Let H be a MD hash function with compression function F. Let a collision for H be given. Then one has either a collision for F, or a preimage of H_0 for F (Fixed Point).

Theorem 4

Let H be a MD hash function with compression function F and with MD-strengthening. Let a collision for H be given. Then one has found a collision for F.

Some Disadvantages of Merkle-Damgård

- “Length-Extension” ($\rightarrow$ blackboard)
- And related structural weaknesses (soon to follow)

MD4 (“Message Digest 4”)

- Rivest, 1990
- Compression function with three internal rounds, each round with 16 internal steps, 128-bit output
- Since 1993 not much used in practice (successor: MD5)

- collisions
  - for MD4 reduced to 2 rounds (Merkle, 1990, Bosselaers, den Boer, 1991)
  - all 3 rounds (Dobbertin, 1996), 2^{20} *Ops*
  - all 3 rounds (Wang et al., 2004) *by hand*
- (Second) Preimages for 2 rounds (Dobbertin, 1997)

MD5 (“Message Digest 5”)

- Rivest, 1991
- Compression function with four internal rounds, each round with 16 internal steps, 128-bit output

- Collisions
  - For compression function but not for hash function (Dobbertin, 1996)
  - For hash function (2^{39} Ops, Wang et al., 2004)
  - For hash function (a few seconds on a PC)

Development of Security:

- 1992: Some experts from the European RIPE-project advise against MD5
- 1996: All experts advise against MD5;
  this advice has been greatly ignored by the industry
- 2005: With great hectic, people move away from MD5,
  still not completely abandoned

SHA-1 (“Secure Hash Algorithm One”)

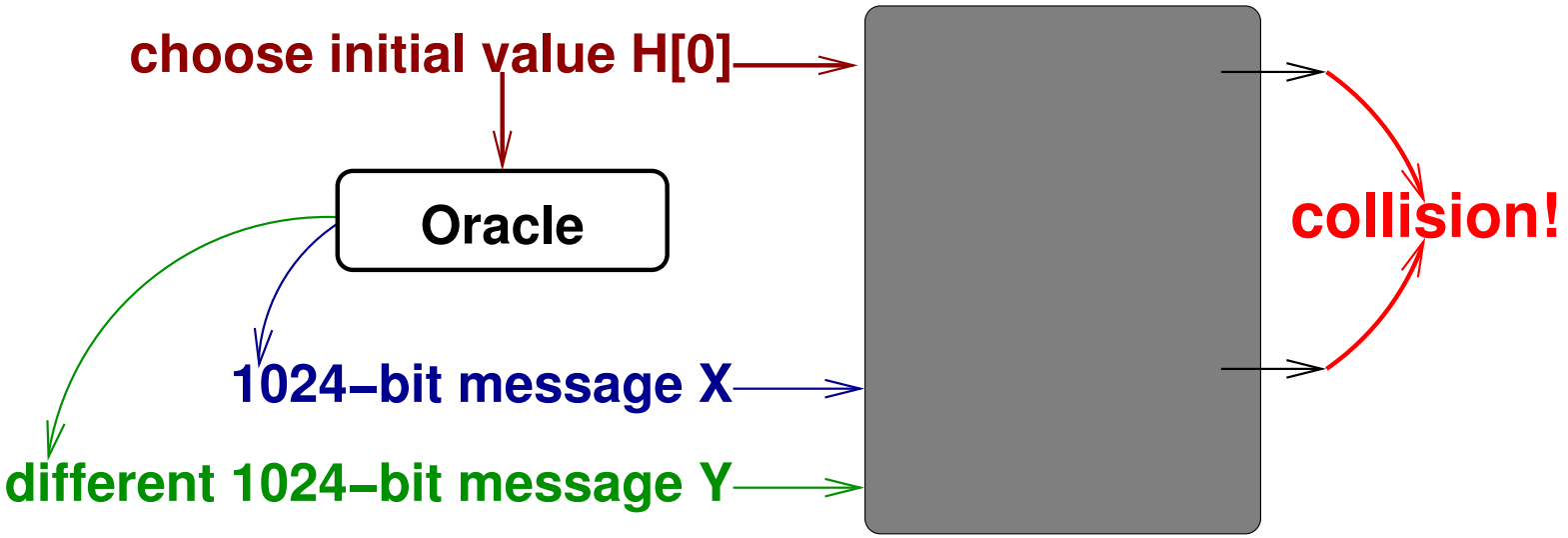
- 1993: SHA (later “SHA-0”), developed by the NIST and the NSA
- 1995: SHA-1 (minor change), also NIST/NSA
- 5 internal rounds, 160 bits of output

- 2004: First attacks against SHA-0 (Joux et al., 2^{51} Ops.)
- 2005: Improved attacks against SHA-0 (Wang et al., 2^{39} Ops.)

- 2005: First attacks against SHA-1 (Wang et al., claim 2^{69} Ops.)
- So far, attacks have not been implemented / demonstrated

Structural Weaknesses of M-D

- Example: the Wang/Yu attack against MD5 . . .



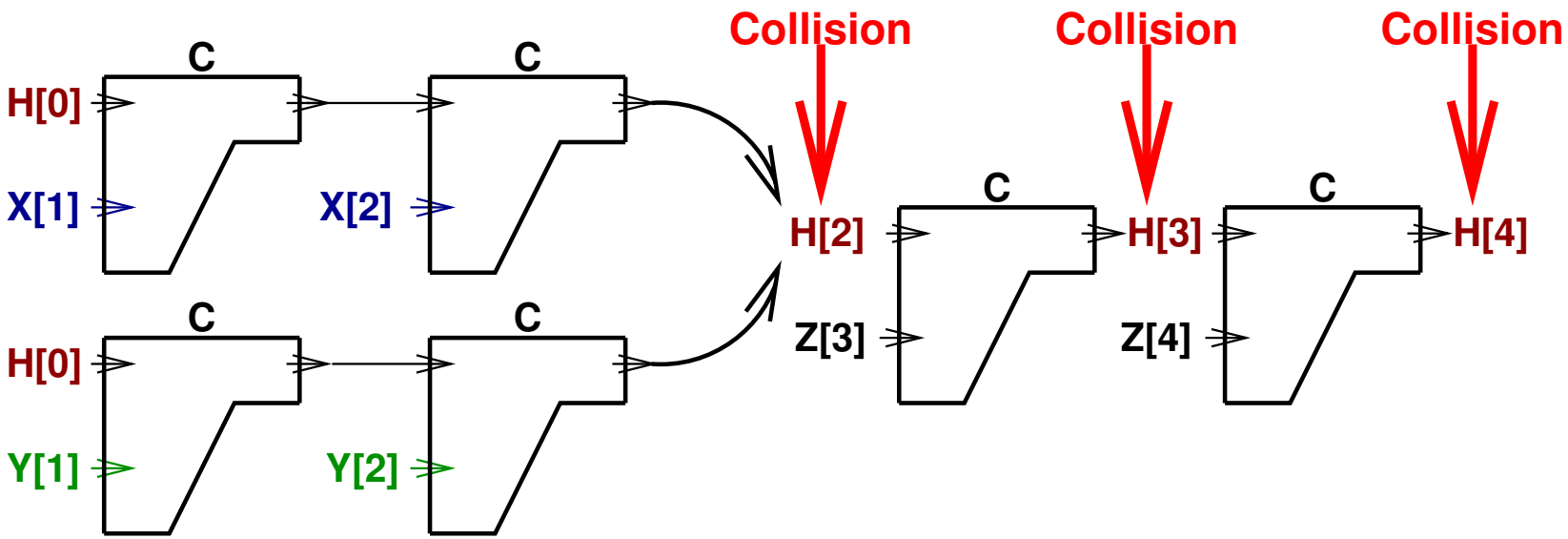
. . . provides „**random“ Collisions** (**X**[**1**], **X**[**2**]) and (**Y**[**1**], **Y**[**2**]).

Random collisions should not be too bad

“Second preimages, yes, these would be damaging!”

But . . .

Colliding messages can be extended to form **new collisions**:



When we start with a random collision, can we do something to turn it into a “useful” one?

The exploit (Daum, Lucks, 2005)

0. Alice: internship in Rome.
1. Letter of recommendation (LOR); on paper
2. Sends a file with the LOR and asks for a digital signature
3. Caesar compares his LOR to the file and then signs it
4. When Caesar is gone to Goal, Alice presents a digitally signed order which grants her access to secret documents . . .

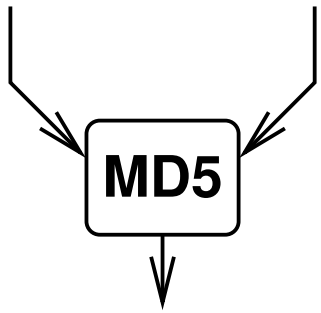
What did Alice do?
Finding a Second Preimage

letter

. . . I highly recommend hiring her.

order

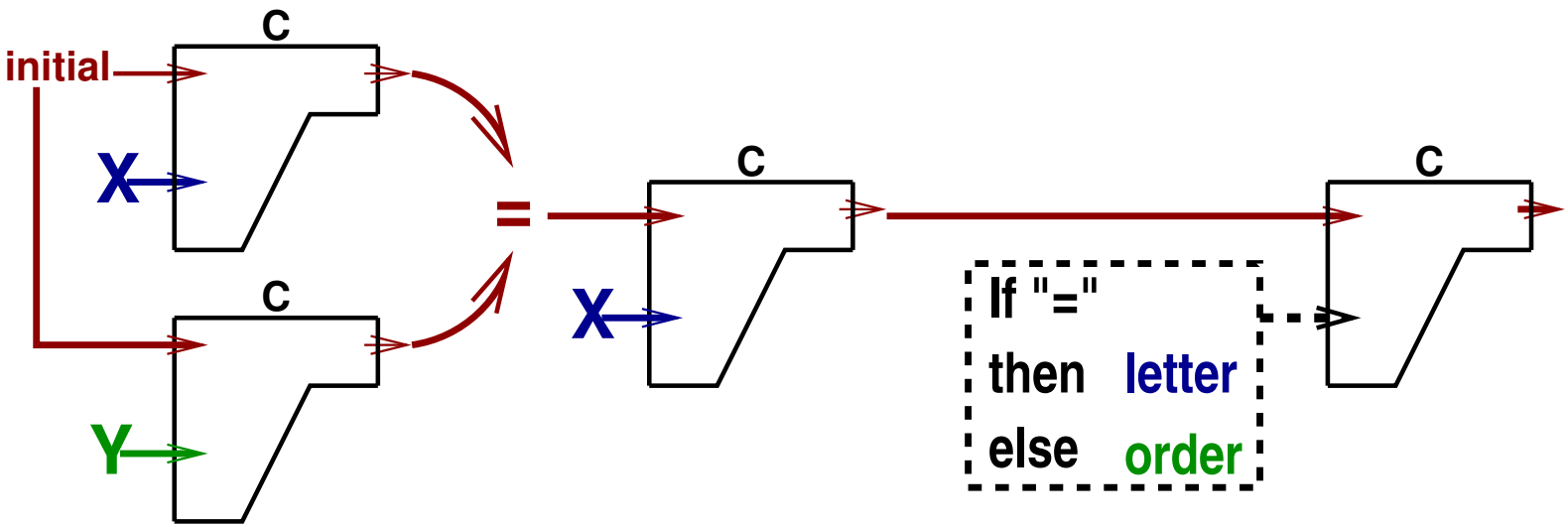
. . . full access to all . . . secret information . . .



5421a523481fdc6a2a1c832e72c7b8a5

No second preimage, only “random” collisions:

- random **X** $\neq$ **Y** with **H(X)**=**H(Y)** (Wang/Yu).
- Postscript-Code:
  put **X** or **Y** onto stack
- put **X** onto stack
- if two entries on stack are **equal**, then print **letter** else print **order**.



Two Postscript Documents

This shows the Principle, Not the PostScript Syntax

$$\text{preamble; } \texttt{put}(\mathbf{X}); \underbrace{\texttt{put}(\mathbf{X}); \texttt{if}(=) \texttt{ then } T_1 \texttt{ else } T_2;}_{=}$$

$$\text{preamble; } \texttt{put}(\mathbf{Y}); \overbrace{\texttt{put}(\mathbf{X}); \texttt{if}(=) \texttt{ then } T_1 \texttt{ else } T_2;}^{=}$$

Similar attacks

- For documents *without* if-then-else (PDF, pictures, . . .) (Gebhard, Illies, Schindler)
- For executables (Kaminski, Mikle)
- For RSA-keys and certificates (Lenstra, Stevens, Wang, de Weger)

Other Exploits of the Merkle-Damgård Structure

Once Finding Any Collision Has Become Feasible

- Joux (2004): Multi-collisions and attacks against hash cascades

- The Nostradamus attack (Kelsey, Kohno, 2006)

2.1: Indifferentiability
Iterated Hash Functions "Without" Structural Weaknesses

Indifferentiability from a Random Oracle (Coron, Dodis, Mailinaud, Puniya, 2005)

The adversary has a pair of oracles (FIL, VIL),

FIL (Fixed input length) – the compr. function

VIL (Variable input length) – the hash function

"real"

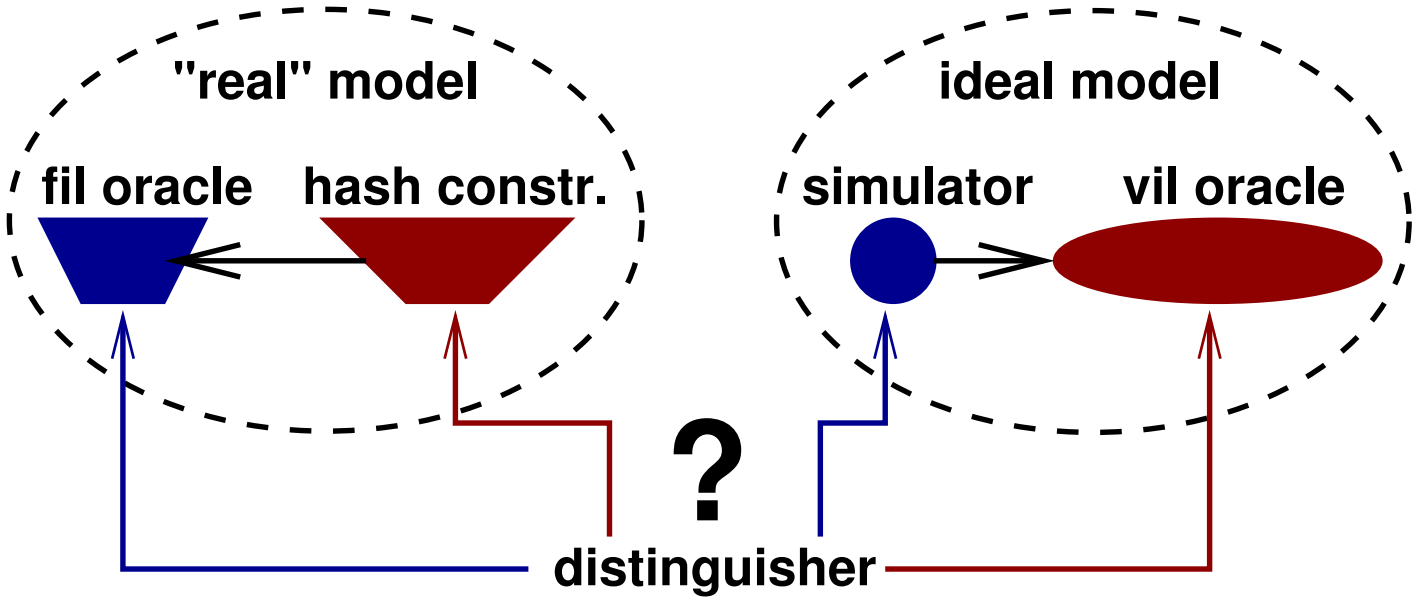
1. The FIL random oracle **F**
2. The VIL constr. $\mathbf{H_F}$

ideal

1. Some fil simulator **S**
2. The VIL random oracle **H**

and tries to distinguish "real" from ideal.

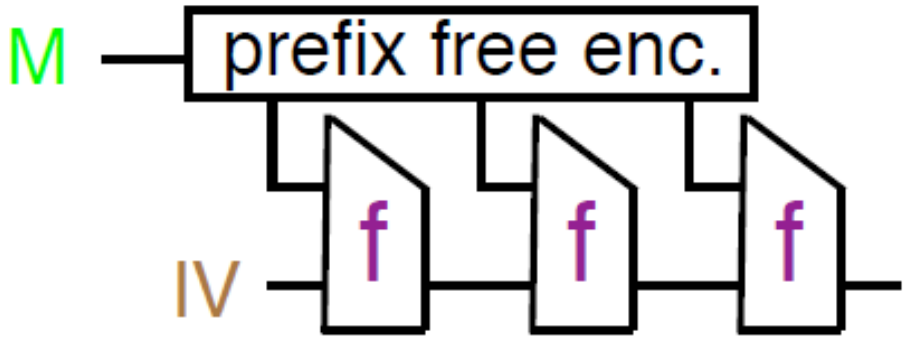
The construction $\mathbf{H_F}$ is good ("indifferentiable from a random oracle"), if an efficient simulator **S** exists, such that it is hard for all adversaries to distinguish "real" from ideal.



Negative examples:

- Plain Merkle-Damgård
- The two-level construction $\mathbf{H}_{\mathbf{F}}(x) = \mathbf{F}(h(x))$ with a collision-resistant h and a fil random oracle $\mathbf{F} : \{0,1\}^n \to \{0,1\}^n$ (we will revisit this construction some time later)

Prefix-Free MD and Its Structure



- Encode input; no M is a prefix of another (longer) M' i.e., there is no X with $M' = (M||X)$
- E.g., append 0 to each internal block and 1 to the last
- This foils extension attack because output of $\mathbf{H_F}$ does not reveal the entire internal state
- Is p-f MD indifferentiable from a RO?

Theorem 5

Prefix-Free MD is indifferentiable from a random oracle.

Core ideas for the proof:

- Simulator **S** distinguishes two query types
  - Internal query (H_{i-1}, M_i):
    - Give random answer
    - Keep track of chains $H_0 \to H_1 \to \cdots \to H_i$
  - Final-block query (H_{i-1}, M_i):
    - **If** chain $H_0 \to \cdots \to H_{i-1}$
      Then **H**$(M_1, \ldots M_{i-1}, M_i)$ for nonrandom part of answer
      Else give completely random answer
- As long as there is at most one single chain for a final query, **S** can give an answer consistent with **H**
- Random responses prevent collisions between chains (for $\ll 2^{n/2}$)

Is “Indifferentiable from ROM” = “No Structural Weakness”

- Not really!
- Examples:
  - Reconsider the two-level construction

$$\mathbf{H_F}(x) = \mathbf{F}(h(x)).$$

    Is it really a bad idea? (Homework)
  - Another issue: prove of storage $H(M||C)$ vs. $H(C||M)$
    - H: Iterated hash function
    - M: Large data stored on server (external)
    - C: Challenge from client

    (Homework)
- But at this point of time, cryptographic theory has no better indication for a hash function not suffering from structural weaknesses – finding a better notion is an open research problem.