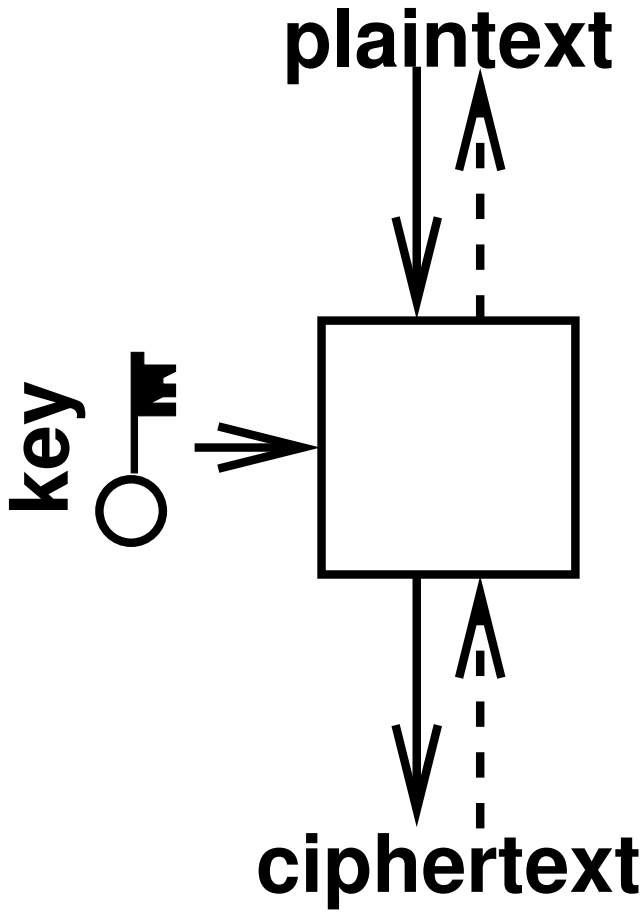
4: Block-Cipher-Based Compression Functions

- block cipher:
  - k-bit key
  - n-bit plain- and ciphertexts
  - invertible for a given key
- naive approach to turn a block cipher into a compression function:
  - n-bit chaining value
  - k-bit message blocks
  - but invertibility is an issue ...



Notions, Advantages and Disadvantages

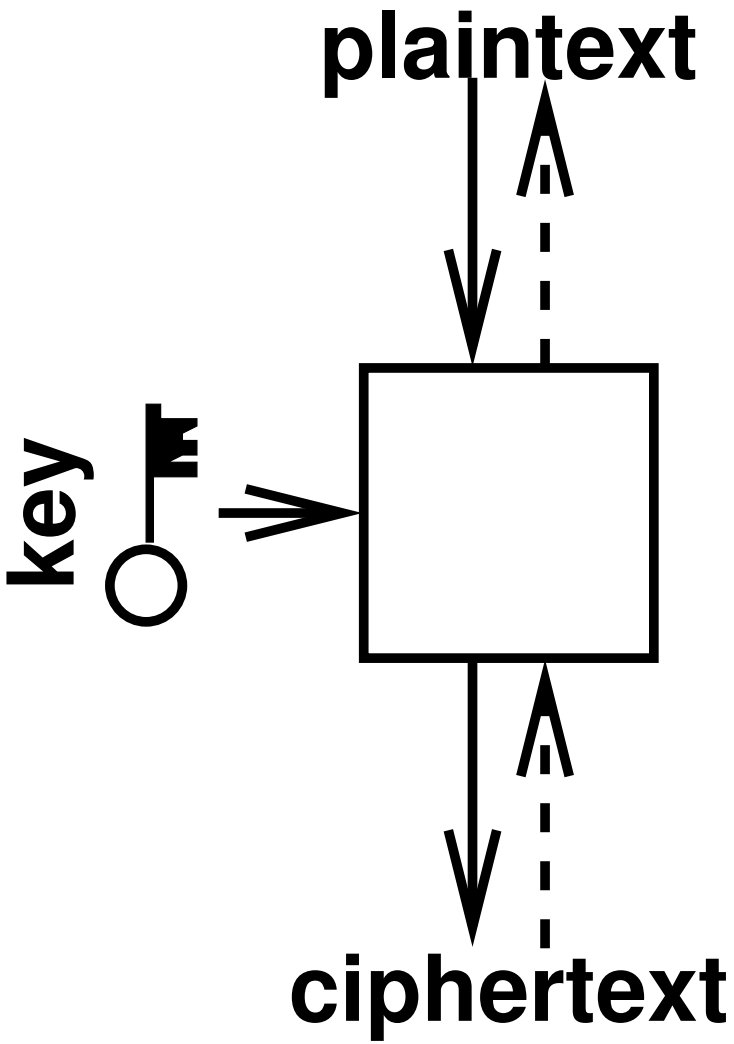
the "rate" of a hash function using an n-bit block cipher is

$$\frac{\#\ n\text{-bit message blocks}}{\#\ \text{calls to block cipher}}$$

for message lengths converging to ∞

+ we (seem to) have some secure and efficient block ciphers
+ many applications need both (bc and hf) – save on code size or chip space
+ well-understood theory
– the block cipher is (mis-)used outside its original specification
– block size n often too small for collision resistance, e.g., $n = 128$
– for such small n, we have to develop a special theory for "Double Block Length" ($2n$-bit) hash functions

Diffie, Hellman, 1976

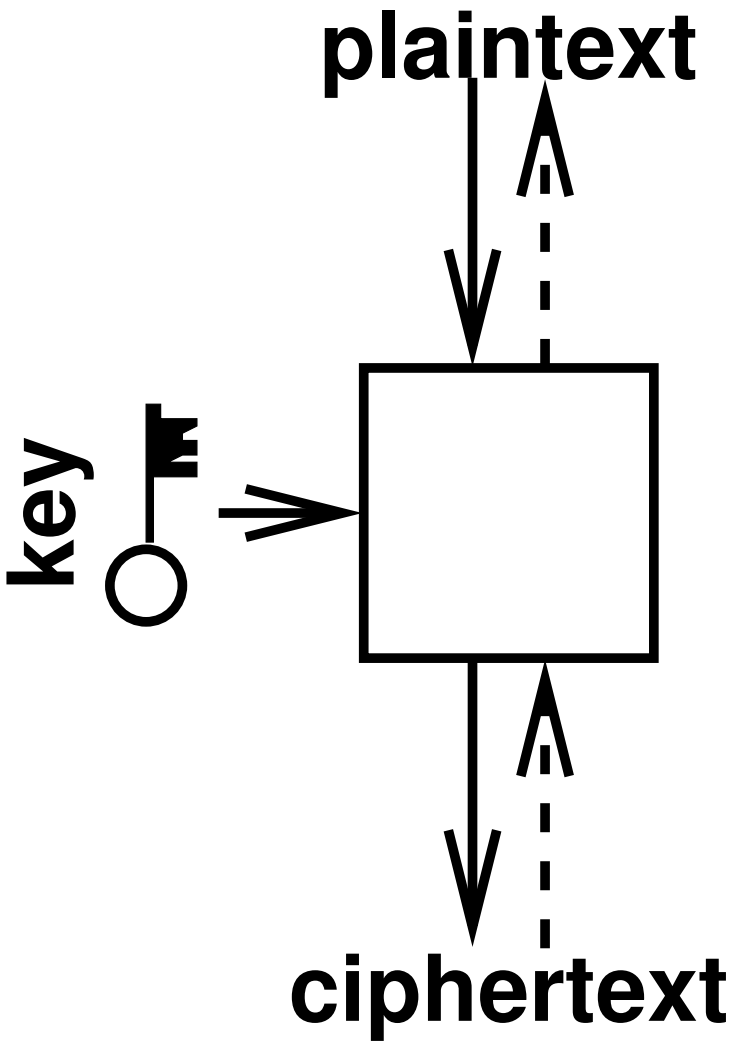


- block cipher
  - k-bit key
  - $(k-n)$-bit plaintext for $k > n$
- compression function
  - n-bit chaining value
  - message block $\mathbf{M_i}$

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}},\mathbf{M_i}}(\text{Const})$$

- If the bc E is secure in the usual sense, then the compression function seems to be secure against pure and second preimages

Rabin, 1978



- block cipher
  - k-bit key
  - n-bit plaintext
- compression function
  - n-bit chaining value
  - k-bit message block $\mathbf{M_i}$

$$\mathbf{H_i} := E_{\mathbf{M_i}}(\mathbf{H_{i-1}})$$

- Yuval: collisions and preimages in $\Theta(2^{n/2})$ (how?)

The “Classical Ones”

- Davies-Meyer:

$$\mathbf{H_i} := E_{\mathbf{M_i}}(\mathbf{H_{i-1}}) \oplus \mathbf{H_{i-1}}$$

- Matyas-Meyer-Oseas:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i}$$

- Miyaguchi-Preneel:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i} \oplus \mathbf{H_{i-1}}$$

4.1: Meet-In-The-Middle (MITM) Attacks

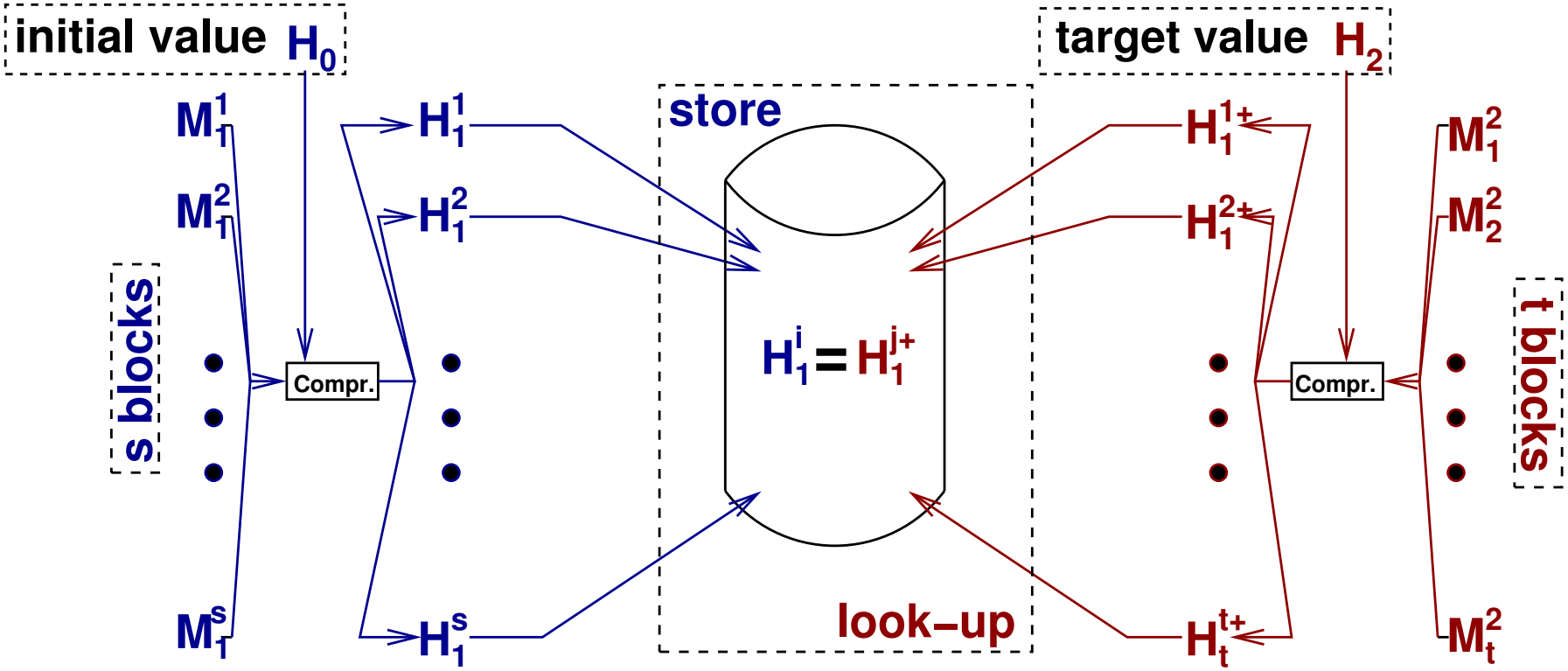
- a general iterated n-bit hash function:
  - initial value $H_0 \in \{0,1\}^n$,
  - iteration $H_i := h(H_{i-1}, M_i)$,
  - final result $H(M_1, \ldots, M_L) = H_L$.
- in this section: *invertible h*, i.e., given H_i and M_i
  - $H_{i-1} := h^{-1}(H_i, M_i)$ is well-defined, and
  - one can compute H_{i-1} efficiently

A MITM Attack

- wanted: preimage for $\mathbf{H}_{\text{Target}}$
- try many $\mathbf{M_1}$ and compute $\mathbf{h}(\mathbf{H_0}, \mathbf{M_1})$
- try many $\mathbf{M_2}$ and compute $\mathbf{h}^{-1}(\mathbf{H}_{\text{Target}}, \mathbf{M_2})$.
- If $\mathbf{h}(\mathbf{H_0}, \mathbf{M_1}) = \mathbf{h}^{-1}(\mathbf{H}_{\text{Target}}, \mathbf{M_2})$,
  then $H(\mathbf{M_1}, \mathbf{M_2}) = \mathbf{H}_{\text{Target}} \longrightarrow$ preimage found!
- naive implementation: time and storage $\Theta(2^{n/2})$.

- thus: n-bit hash functions with invertible n-bit compression functions in general are as vulnerable against preimage attacks as they are against collision attacks

Naive MITM Attacks



- compute s values $H_1^i = h(H_0, M_1^i)$
- compute t values $H_1^{j+} = h^{-1}(H_2, M_2^j)$ wanted: i, j: $H_1^i = H_1^{j+}$.
- condition: $st = 2^n$. time: $s + t$. memory: $\min\{s, t\}$.

Memory-Efficient MITM Attacks

- previous collision attacks:
  - one function f
  - storage $\Theta(1)$ instead of $\Theta(2^{n/2})$ at the same (asymptotical) time,
  - used cycle finding, distinguished points, ... to search for

$$x \neq y \text{ with } f(x) = f(y) \quad (*)$$

- now, we search for a different type of collisions
  - two different functions $f(x) = h(H_0, x)$ and $g(y) = h^{-1}(H_{\text{target}}, y)$
  - search for

$$x, y : f(x) = g(y),$$

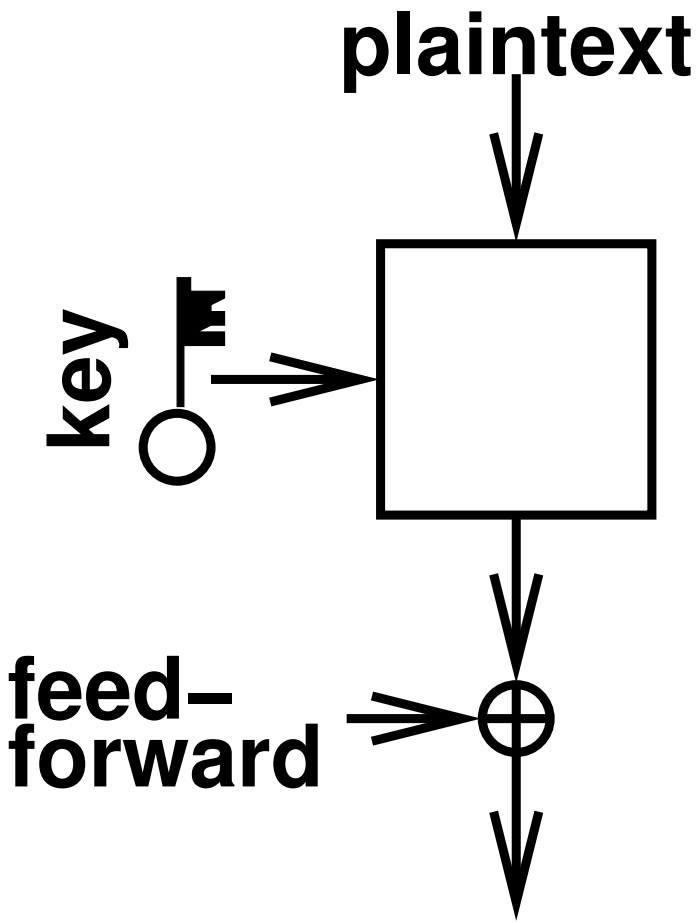
    regardless of $x = y$ or$x \neq y$
  - the idea: modify that problem into a single-function problem of type $(*)$ (but how?)

4.2: The Classification from Preneel, Govaerts and Vanderwalle (1993)

- inspired by the “Classical Ones”
- generalized scheme, including the “Classical Ones”
- search for attacks **independent from the block cipher**, i.e., model the block cipher as an oracle:
  - given inputs X and K, output $Y = E_K(X)$!
  - given inputs Y and K, output $X = E_K^{-1}(Y)$!

PGV: The approach

- Three input slots for the "extended" encryption:
  - plaintext,
  - ciphertext, and
  - feed-forward
- Each input slot may take either of the following values:
  - a constant (w.l.o.g., 0),
  - the chaining value H_{i-1},
  - the message block M_i,
  - or their sum $H_{i-1} \oplus M_i$
- all in all $4^3 = 64$ different compression functions



Attack Classes (sorted by relevance)

1. trivial
2. direct
3. permutation
4. forward
5. (backward)
6. (fixed point)

The first four classes are deadly for any hash function.

Being vulnerable against backward or fixed point attacks is an issue, but may be acceptable in many circumstances.

PGV: Attack Classes 1 and 2

trivial: $\mathbf{H_i}$ can be computed *without* knowing

- both $\mathbf{H_{i-1}}$ and $\mathbf{M_i}$
- or at least $\mathbf{H_{i-1}} \oplus \mathbf{M_i}$

◦ Examples:

$$\mathbf{H_i} := E_0(0) \oplus 0,$$

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(0) \oplus 0,$$

$$\mathbf{H_i} := E_0(\mathbf{M_i}) \oplus \mathbf{M_i}.$$

direct: Given $\mathbf{H_i}$ and $\mathbf{H_{i-1}}$, one can compute $\mathbf{M_i}$ directly.

◦ Example:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(0) \oplus \mathbf{M_i}.$$

PGV: Attack Class 3

permutation: A function f exists with $\mathbf{H_i} = f(\mathbf{M_i}) \oplus \mathbf{H_{i-1}}$.

- Then $\mathbf{H_i} = f(\mathbf{M_i}) \oplus f(\mathbf{M_{i-1}}) \oplus \mathbf{H_{i-2}}$. Why is this bad?
- Since Hash(..., $\mathbf{M_i}$, $\mathbf{M_{i-1}}$, ...) = Hash(..., $\mathbf{M_{i-1}}$, $\mathbf{M_i}$, ...), and collisions and second preimages are easy, then!
- Example:

$$\mathbf{H_i} := E_{\mathbf{M_i}}(\mathbf{M_i}) \oplus \mathbf{M_i} \oplus \mathbf{H_{i-1}}.$$

PGV: Attack Class 4

forward: Given H_{i-1}, H^*_{i-1} and M_i, one can compute M^*_i with

$$\text{Compress}(H_{i-1}, M_i) = \text{Compress}(H^*_{i-1}, M^*_i)$$

without calling the oracle.

- Why is this bad? (Collisions & Second Preimages!)
- Example:

$$H_i := E_{H_{i-1} \oplus M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}.$$

Given M_i, H_{i-1}, and H^*_{i-1}, **how do you compute M^*_i?**
Remember:

$$H_i = E_{H^*_{i-1} \oplus M^*_i}(M^*_i \oplus H^*_{i-1}) \oplus M^*_i \oplus H^*_{i-1}.$$

PGV: Class 5 Weakness

backward: given $\mathbf{H_i}$ it is feasible to find $\mathbf{M_i}$ and $\mathbf{H_{i-1}}$ with

$$\text{Compress}(\mathbf{H_{i-1}}, \mathbf{M_i}) = \mathbf{H_i}.$$

- Pseudo-Preimages, Pseudo-Collisions, MITM-Attacks.
- Example:

$$\mathbf{H_i} := E_{\mathbf{M_i}}(\mathbf{H_{i-1}}) \oplus 0.$$

- We already know that scheme – it is Rabin!
- Another Example:

$$\mathbf{H_i} := E_{\mathbf{M_i} \oplus \mathbf{H_{i-1}}}(\mathbf{M_i} \oplus \mathbf{H_{i-1}}) \oplus \mathbf{H_{i-1}}.$$

PGV: Class 6 Weakness

fixed point: find $\mathbf{H}$ and $\mathbf{M}$ with

$$\mathbf{H} = \text{Compress}(\mathbf{H}, \mathbf{M}).$$

- Is that a problem? Sometimes yes, as you will see soon.
- Example

  $$\mathbf{H_i} := E_{\mathbf{M_i}}(\mathbf{H_{i-1}}) \oplus \mathbf{H_{i-1}}.$$

  Have you seen this scheme before?

  Oh, yes, that is one of the classical ones: Davies-Meyer!

  If fixed points are a weakness, than well-established hash functions (MDx, SHA-1, SHA-2, ...) are suffering.
- Note that you can actually find a fixed point for *every* $\mathbf{M}$! (How?)

PGV: Summary

- systematic classification of 64 possible constructions
- most of them insecure
- *no attacks found:* only four, including
  - Matyas-Meyer-Oseas:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i}$$

  - Miyagucchi-Preneel:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i} \oplus \mathbf{H_{i-1}}$$

- *only fixed points:* another eight, including Davies-Meyer

- an attack-centric approach
- "no attacks found" $\neq$ "there are no attacks"
  "lack of evidence" $\neq$ "proof of absence"

4.3: Black, Rogaway und Shrimpton (2002) – the Proof-Centric Approach

- Model:
  - ideal block cipher
  - # oracle queries, the adversary makes
  - apart from that, the adversary is computationally unbounded
- proofs of security for the four “secure” and the eight “fixed point” constructions
  - collisions require $\Theta(2^{n/2})$ queries
  - preimages require $\Theta(2^n)$ queries

  $\Rightarrow$ Great! PGV did not overlook anything important!
- proofs of security for another eight being vulnerable to “backward” attacks, including Rabin:
  - collisions require $\Theta(2^{n/2})$ queries
  - preimages also require $\Theta(2^{n/2})$ queries
  - this is the best one could hope for:
    (MITM attacks in $\approx 2^{n/2}$ queries, exploiting the “backward” property)

The Ideal-Cipher-Model

- The attacker asks **q** queries, each of either of the following two forms:
  - Given K and x, ask for $y = E_K(x)$.
  - Given K and y, ask for $y = E_K^{-1}(y)$.
- The attacker asks on "redundant" queries:
  - Having asked for $y = E_K(x)$, she will neither repeat that query nor ask for $E_K^{-1}(y)$.
  - Having asked for $x = E_K^{-1}(y)$, ... neither repeat ... nor ask for $E_K(x)$.

Security Proofs in the Ideal Cipher Model

- The adversary **succeeds** in finding a collision, if she finds $(\mathbf{H}, \mathbf{M}) \neq (\mathbf{H}^*, \mathbf{M}^*)$ with

$$\text{Compress}(\mathbf{H}, \mathbf{M}) = \text{Compress}(\mathbf{H}^*, \mathbf{M}^*).$$

- She made all necessary queries to actually compute $\text{Compress}(\mathbf{H}, \mathbf{M})$ and $\text{Compress}(\mathbf{H}^*, \mathbf{M}^*)$.
- Depending on $\mathbf{q}$, we compute (an upper bound for) the probability for the adversary to succeed.

Example: One of the Results from BRS

compression function:

$$\mathbf{H_i} := E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i}$$

collision: $(\mathbf{H_{i-1}}, \mathbf{M_i}) \neq (\mathbf{H^*_{i-1}}, \mathbf{M^*_i})$ with

$$E_{\mathbf{H_{i-1}}}(\mathbf{M_i}) \oplus \mathbf{M_i} = E_{\mathbf{H^*_{i-1}}}(\mathbf{M^*_i}) \oplus \mathbf{M^*_i}$$

Theorem 6

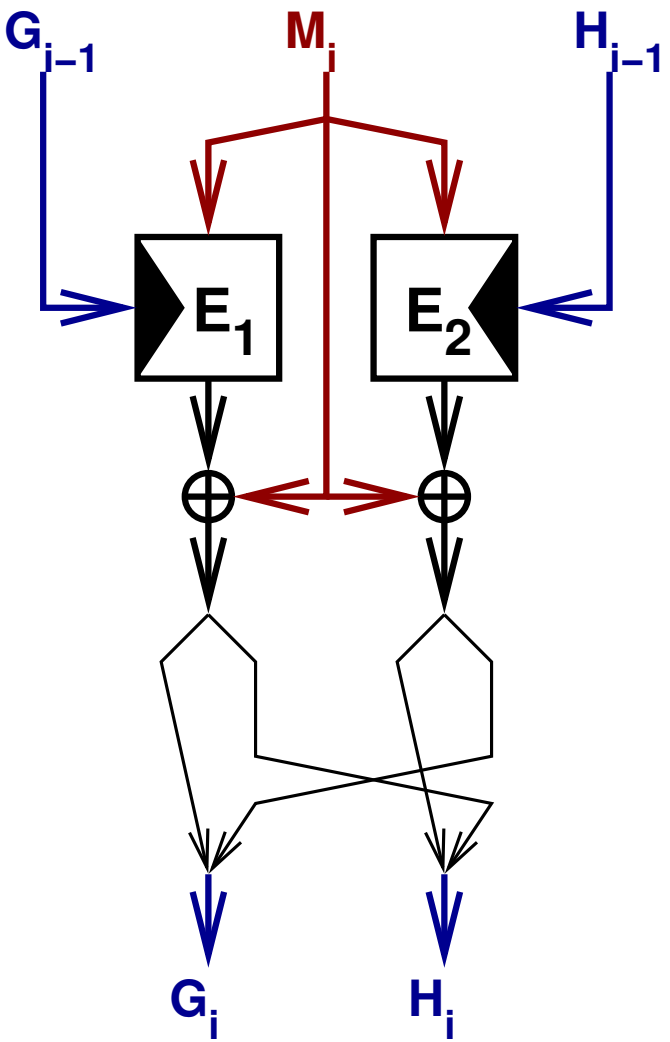
Let $\mathbf{q} > 1$*. The adversary's probability to succeed in finding a collision for the above compression function is at most*

$$\Theta\left(\frac{\mathbf{q}^2}{2^n}\right).$$

4.4: “Double Block Length” Hash functions

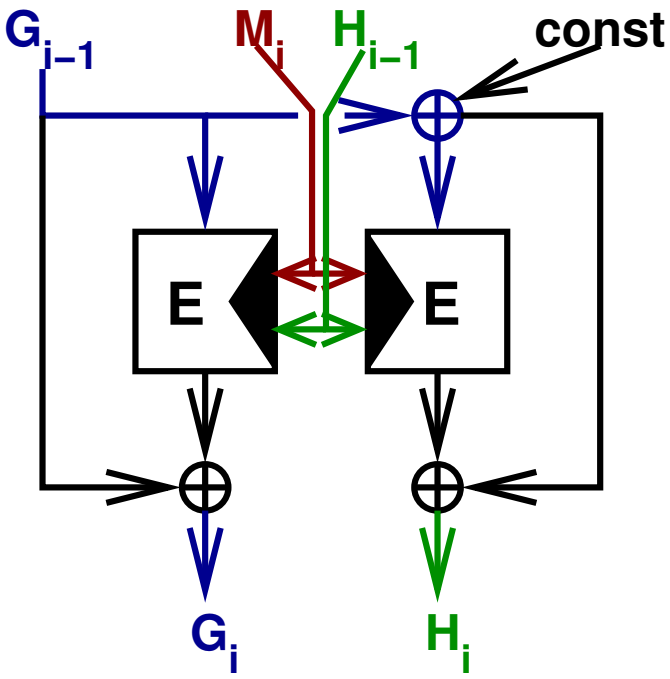
- common block ciphers (e.g., AES):
  - block size $n = 128$
  - key lengths between $n = 128$ and $2n = 256$
- how to construct a $2n$-bit hash function?
- what we would like to have is one rate-1/2 hash function with
  - collision resistance at $\Theta(2^n)$ queries,
  - preimage resistance at $\Theta(2^{2n})$, queries
- a very active research area, with many open problems
- great progress made recently, for the case of $2n$-bit keys

MDC2 (IBM 1987) – Greetings from Practice!



- 2 “slightly different” block ciphers E_1 and E_2
- n-bit blocks and n-bit keys (the difficult case)
- 2 “parallel” compression functions
- swap the right halves of both outputs
- best known attacks (by number of oracle queries)
  - collisions $\approx 2^n$
  - preimages $\approx 2^n$ (storage $*$ queries $\approx 2^{2n}$)
- best proven result (Steinberger, 2007): collisions at least $\Theta(2^{3n/5})$
- still a rather disappointing bound far away from the best attack

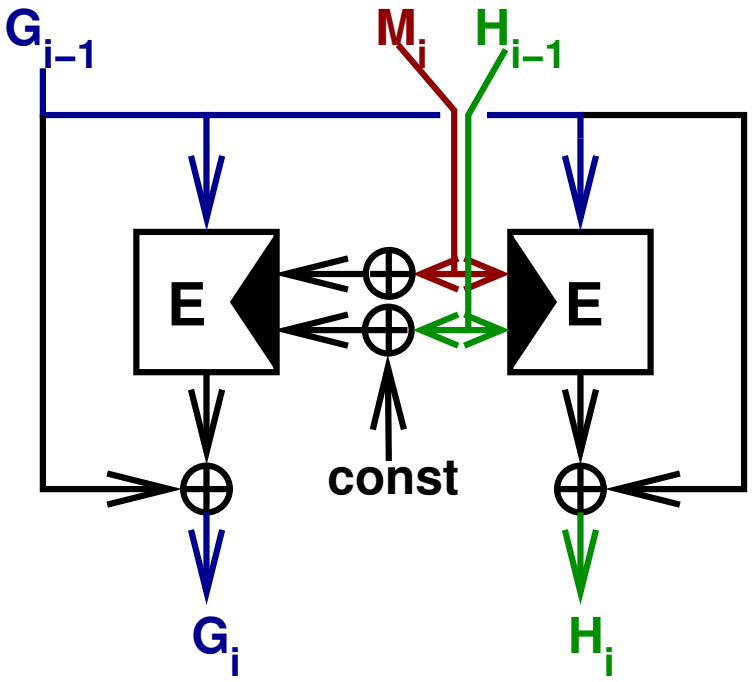
Hirose (2006)



- one n-bit block cipher, $2n$-bit keys
- all $3n$ input bits go into both block cipher calls
- requirement: $\texttt{const} \neq 0$ (why is this important?)
- two block cipher calls with the same key (good for performance)

- proven result: collisions need $\Theta(2^n)$ calls
- concrete for $n = 128$: one needs 2^{125} queries for a success probability 1/2
- proven result: preimages need $\Theta(2^{2n})$, concrete 2^{252} queries

Weimar-DM (2012)



- one n-bit block cipher, $2n$-bit keys
- all $3n$ input bits go into both block cipher calls
- requirement: const$\neq 0$ (why is this important?)
- two block cipher calls with different keys (a bit worse than Hirose)

- proven result: collisions need $\Theta(2^n)$ calls (same as Hirose)
- concrete for $n = 128$: $2^{126.73}$ queries for a success probability 1/2 (This improves over Hirose and is almost optimal!)
- proven result: preimages need $\Theta(2^{2n})$, concrete 2^{252} queries (same as Hirose)