

Hash functions

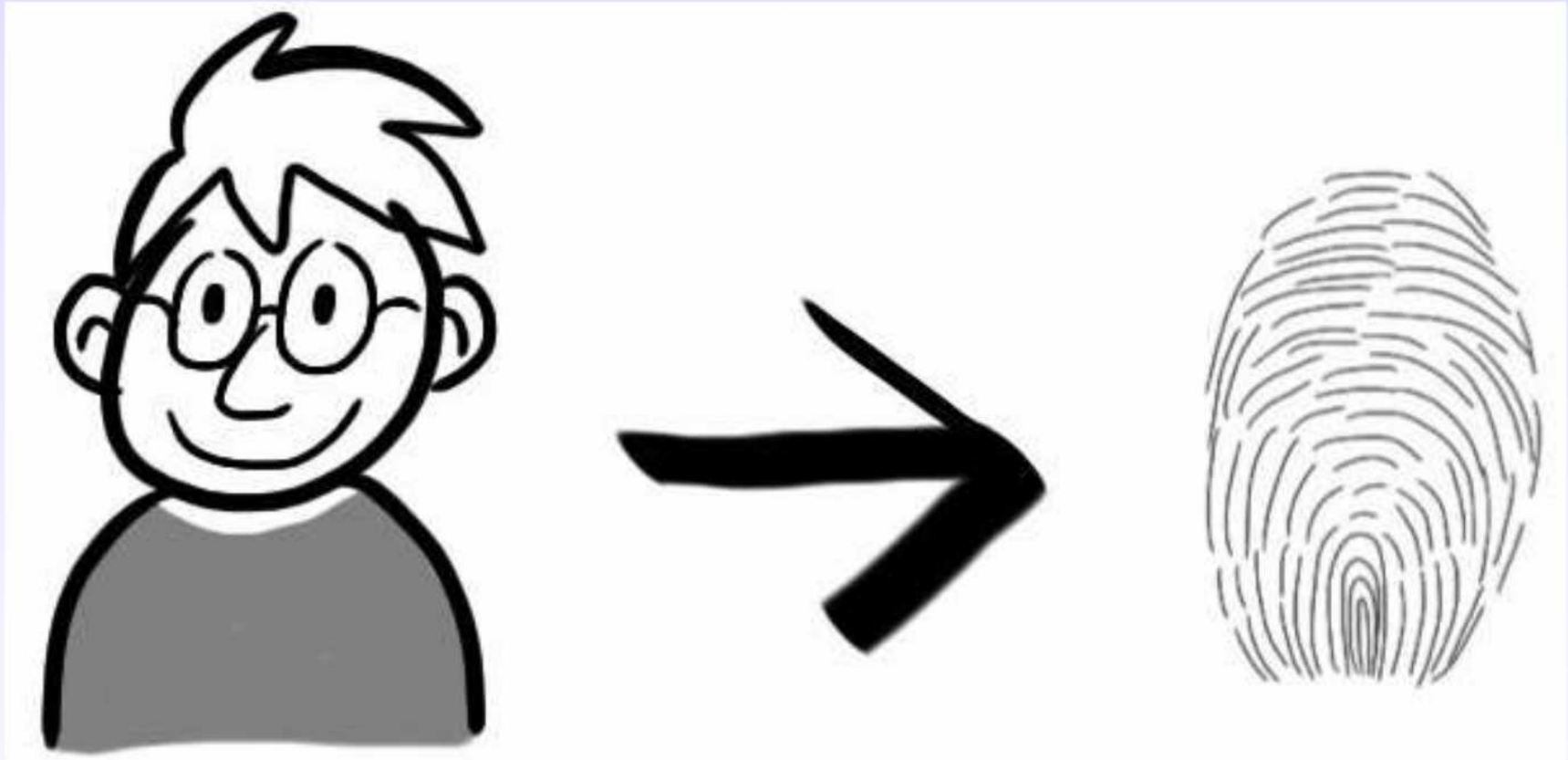
Chanathip Namprempre

Computer Science
Reed College

Agenda: Hash functions

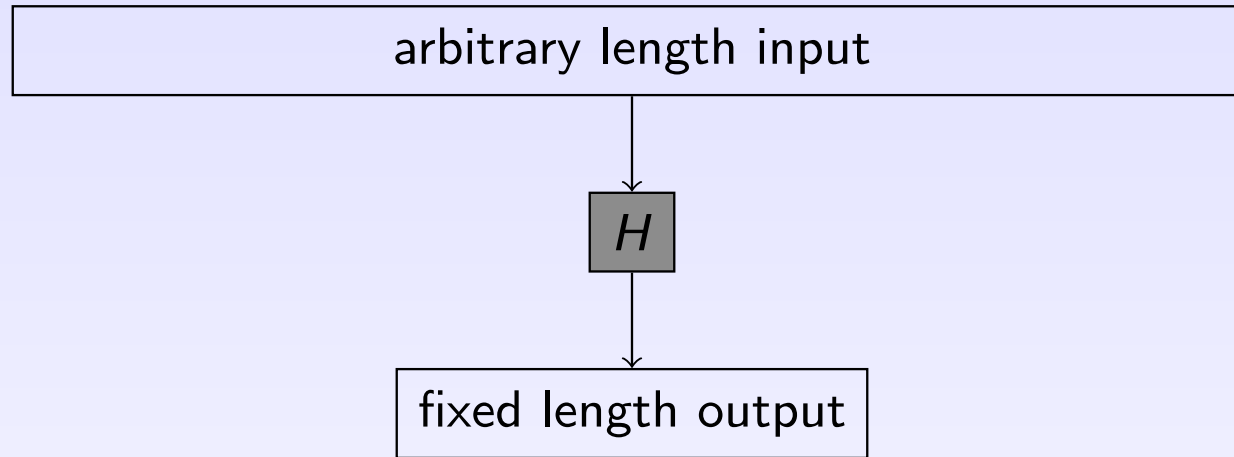
1. Concept
2. Hash function design
3. SHA1
4. Security definitions for hash functions: pre-image attacks, second pre-image attacks, collision attacks
5. Birthday attack
6. Building hash functions from compression functions: MD transform
7. Building compression functions from block cipher: DM transform

A way to think about hash functions



A way to think about hash functions. The digest is supposed to capture the essence of the object being hashed.

Hash functions



The output is often called a **message digest**.

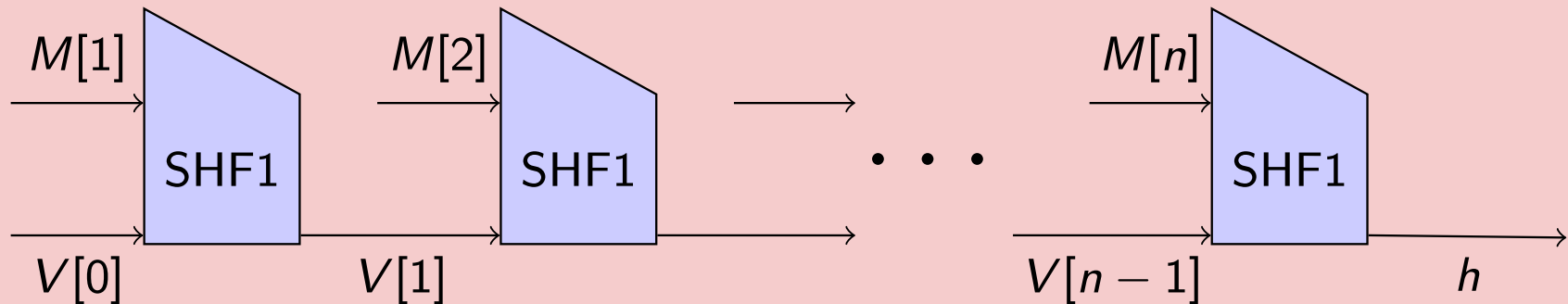
Example: **SHA1** : $\{0, 1\}^{<2^{64}} \rightarrow \{0, 1\}^{160}$

Applications: how hash functions are used

- ▶ Password hashing: good idea, but be careful
- ▶ Message integrity: bad idea
- ▶ Fingerprint of large file: good idea, but make sure the fingerprint is the “real” one.
- ▶ Downline load security: same idea as above
- ▶ Digital signature efficiency: sign $H(M)$ instead of signing M

Hash function design: Merkle-Damgård

Merkle-Damgård: for example SHA1

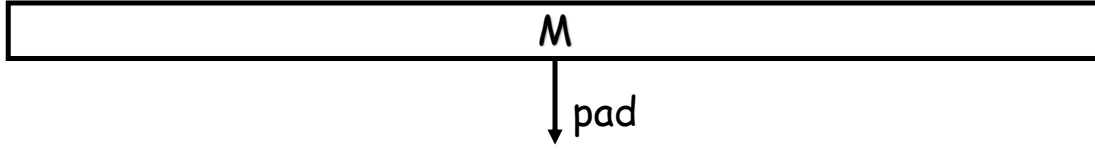


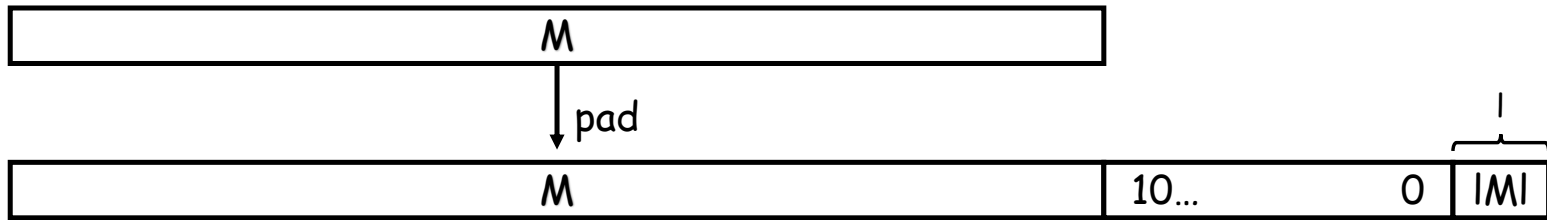
The Merkle-Damgård construction.

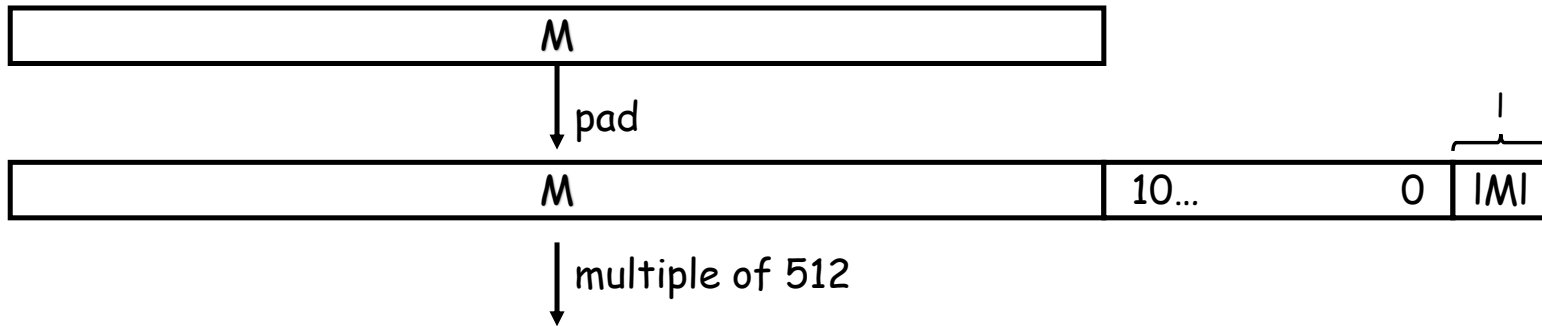
- ▶ The trapezoids represent a compression function.
- ▶ The initial vector $V[0]$ is a fixed, public value.
- ▶ The messages $M[1], \dots, M[n]$ are fixed-length blocks of the padded input message.
- ▶ The output is the output h of the compression function on the last round.

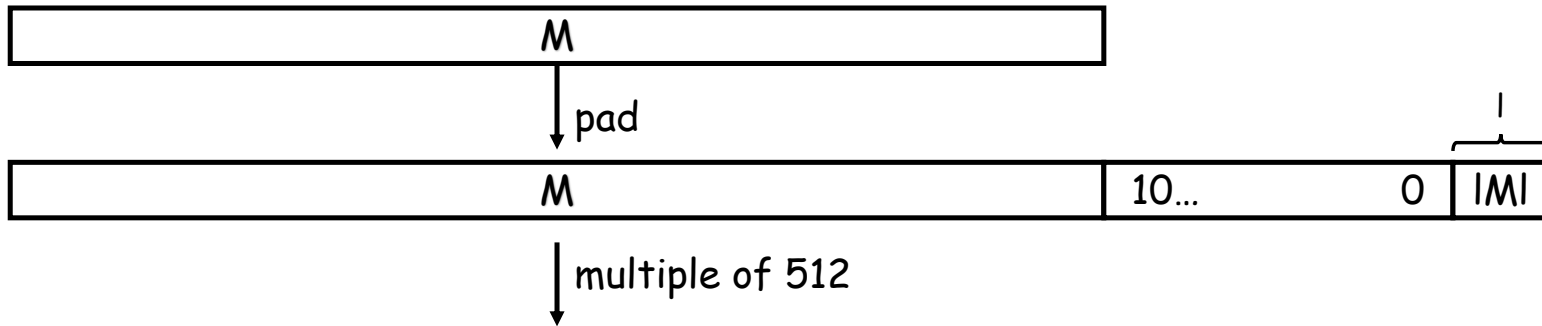
M

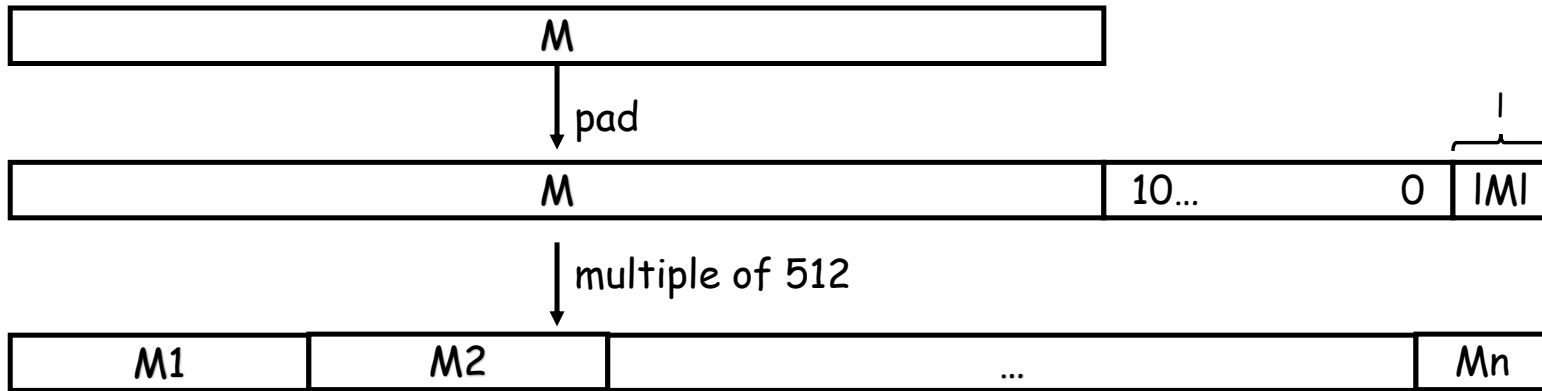


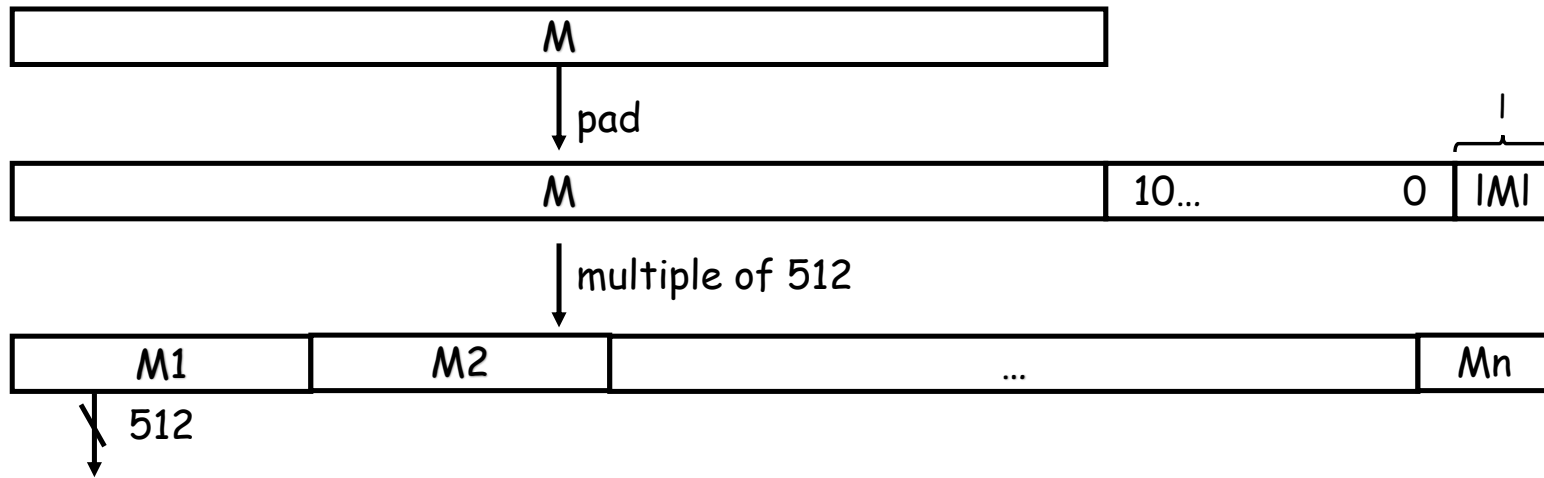


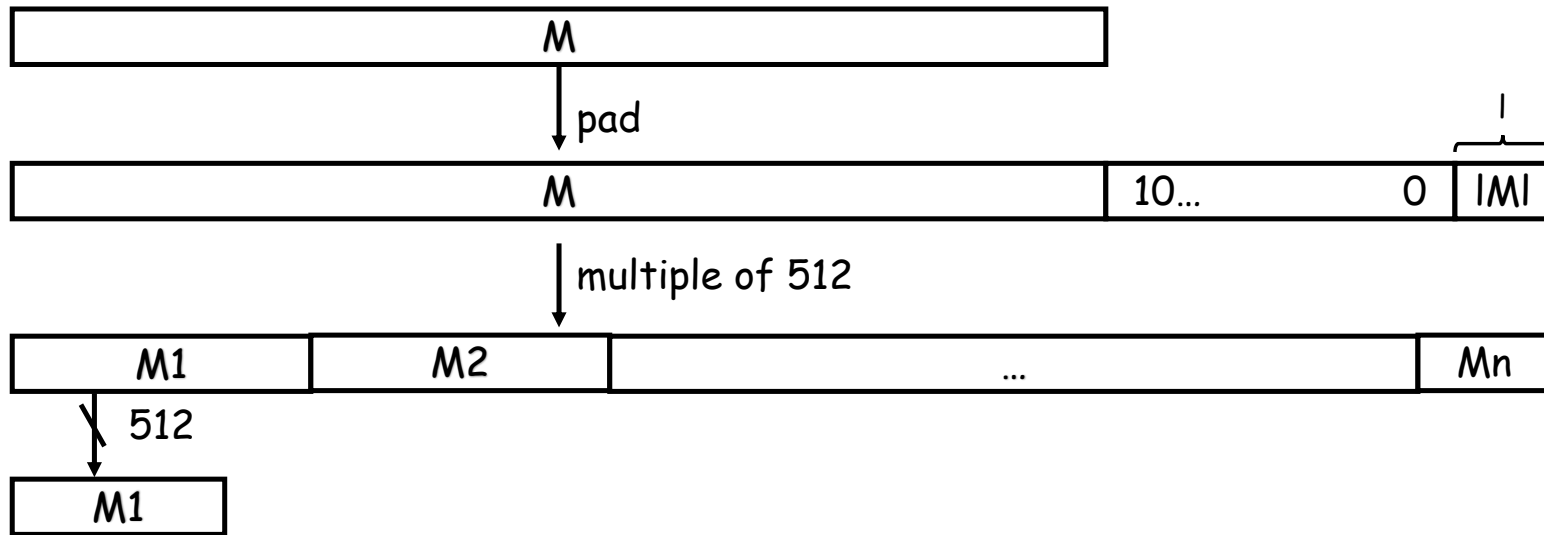


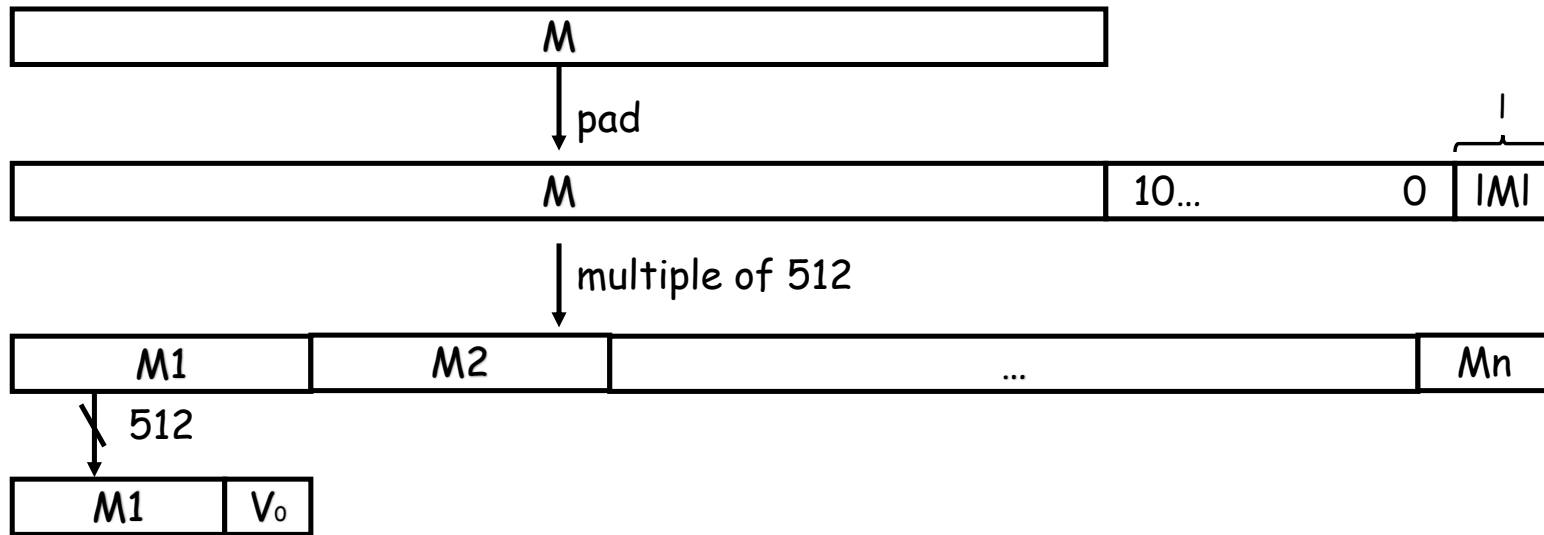


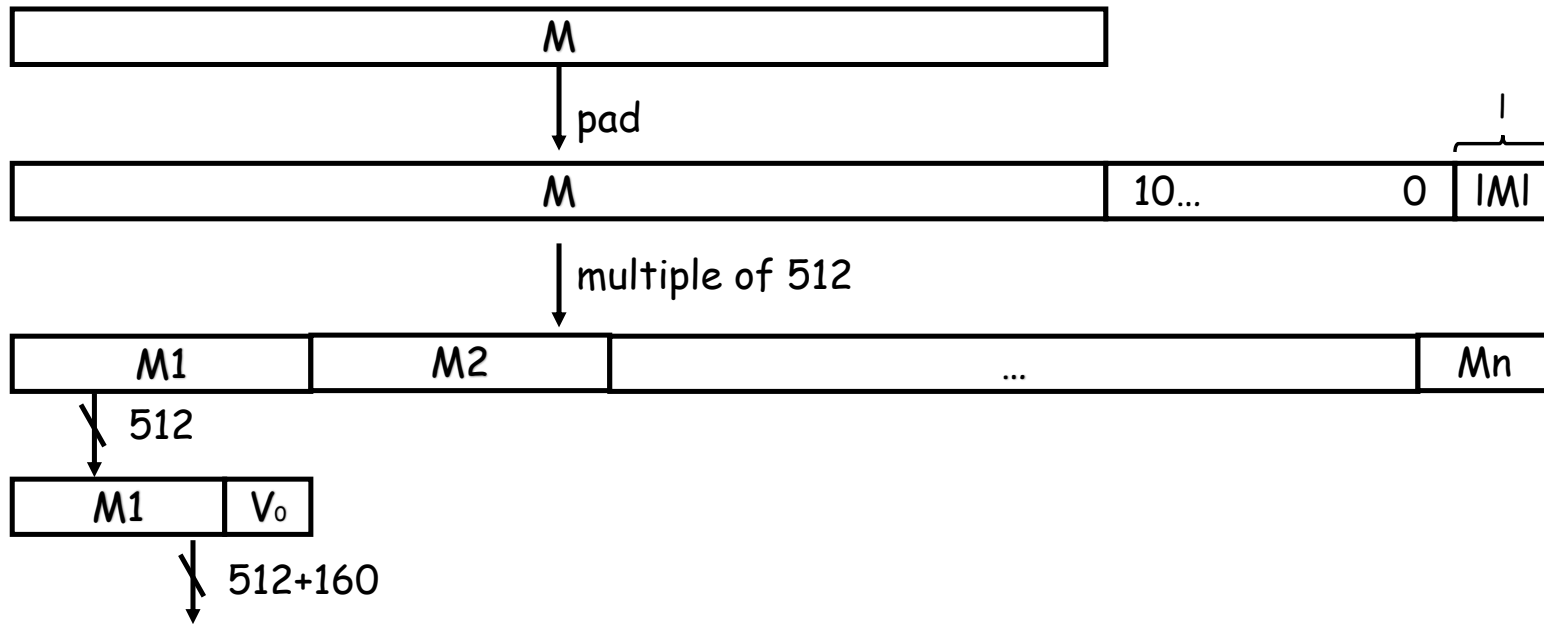


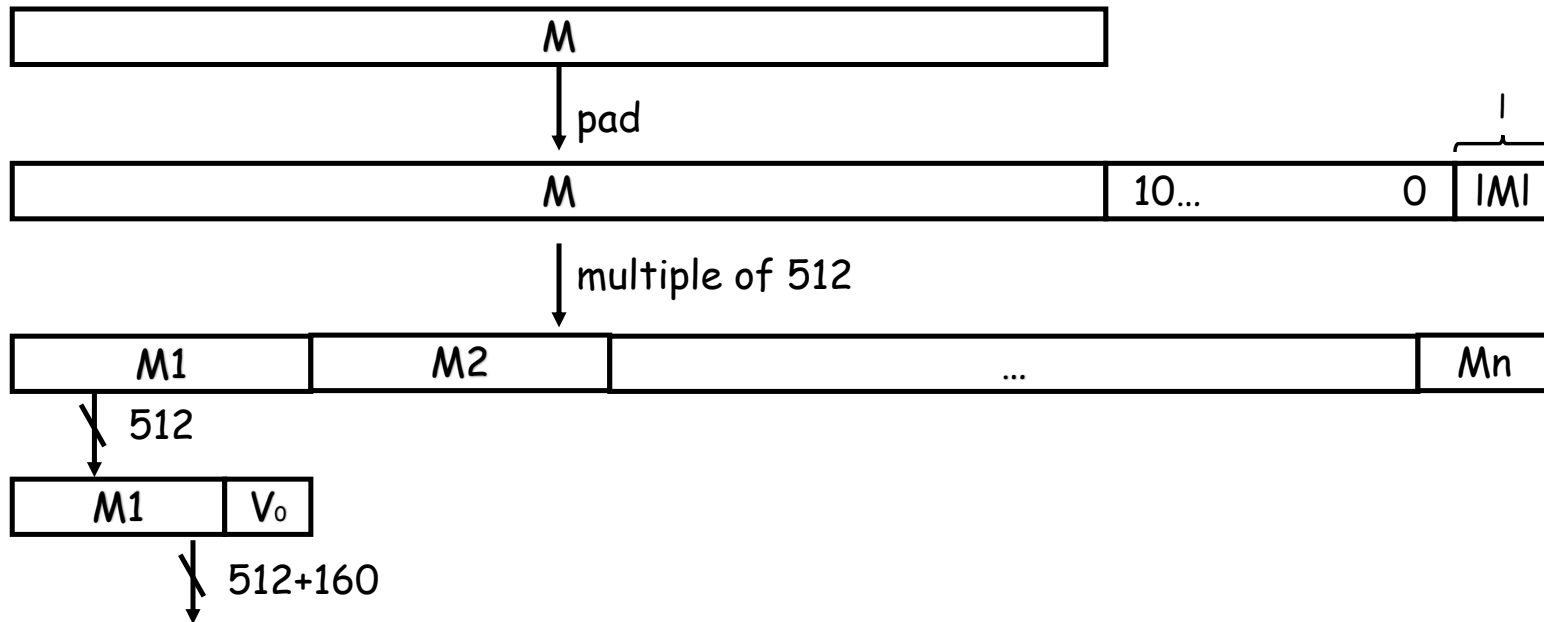






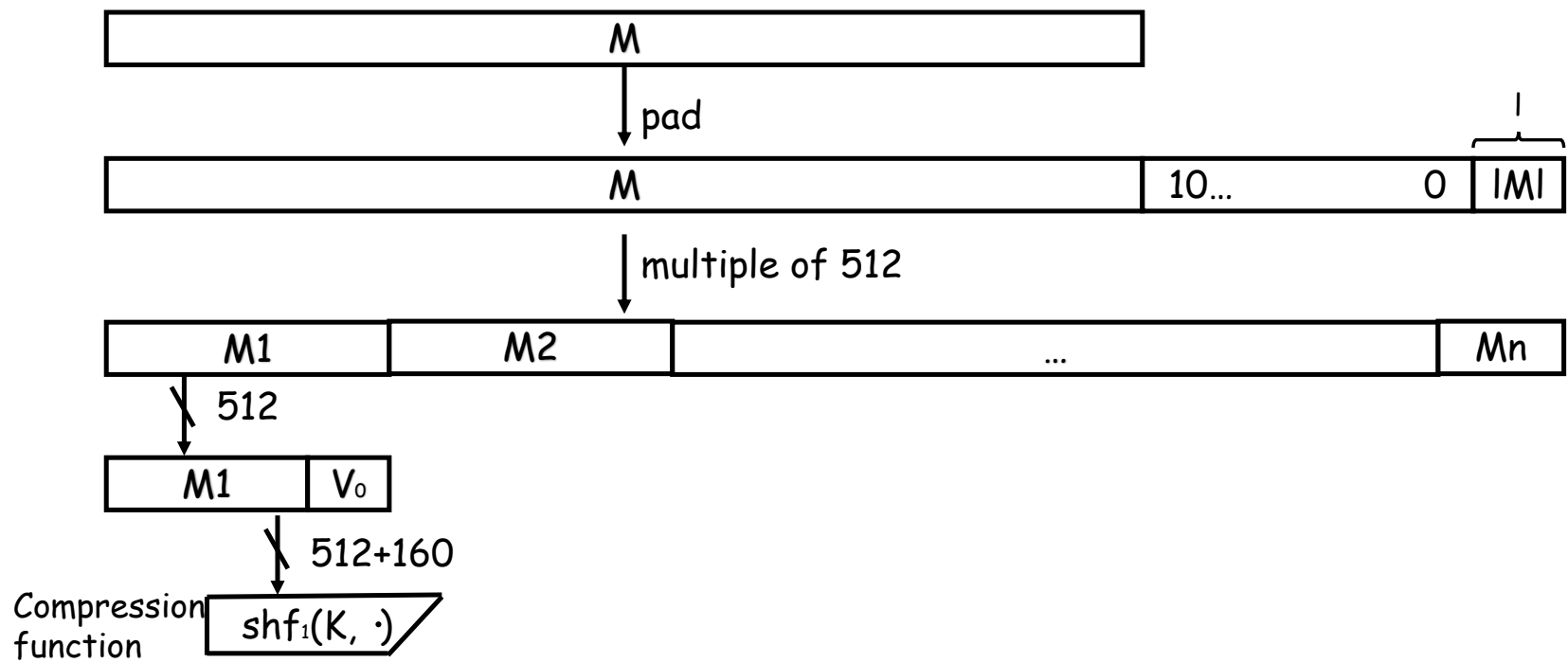


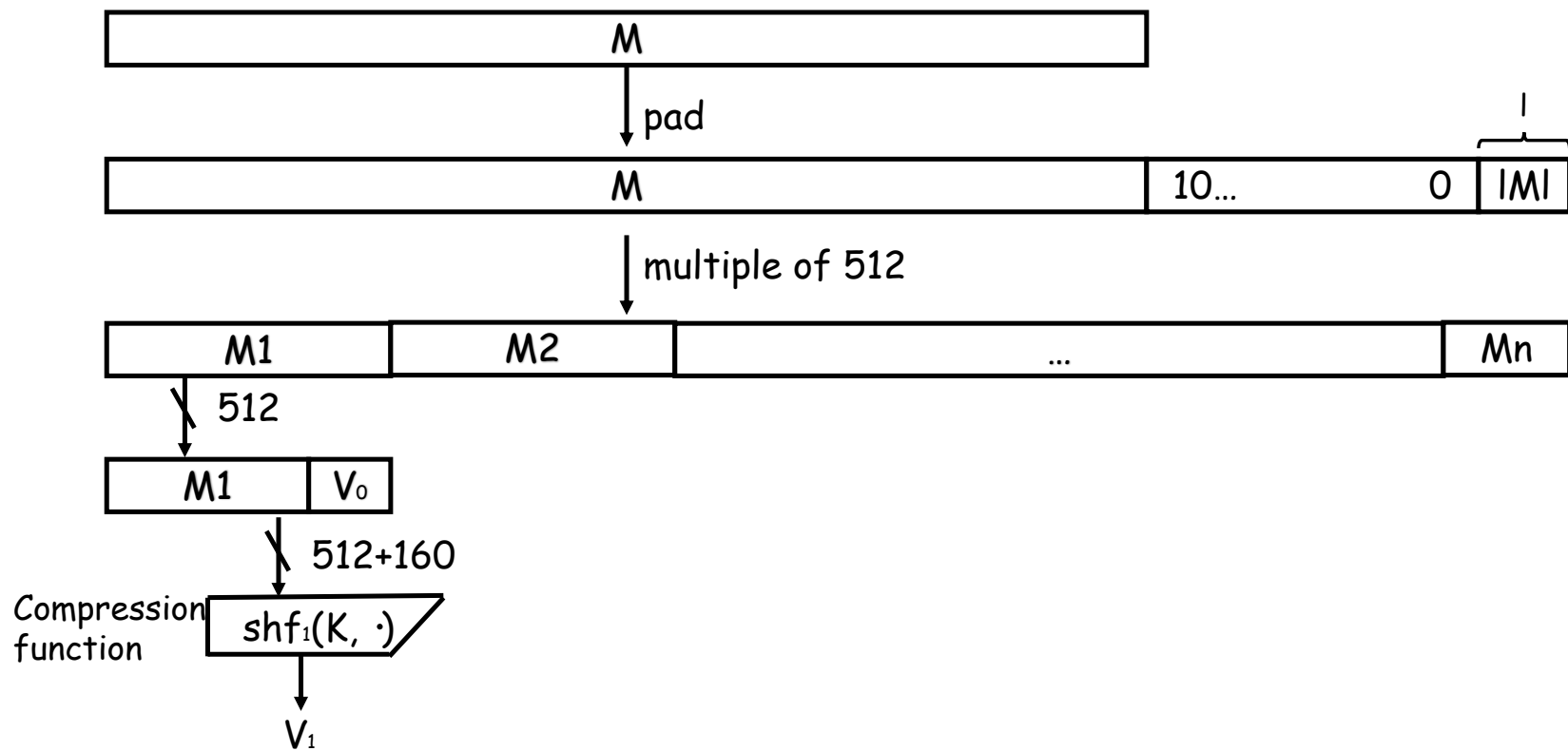


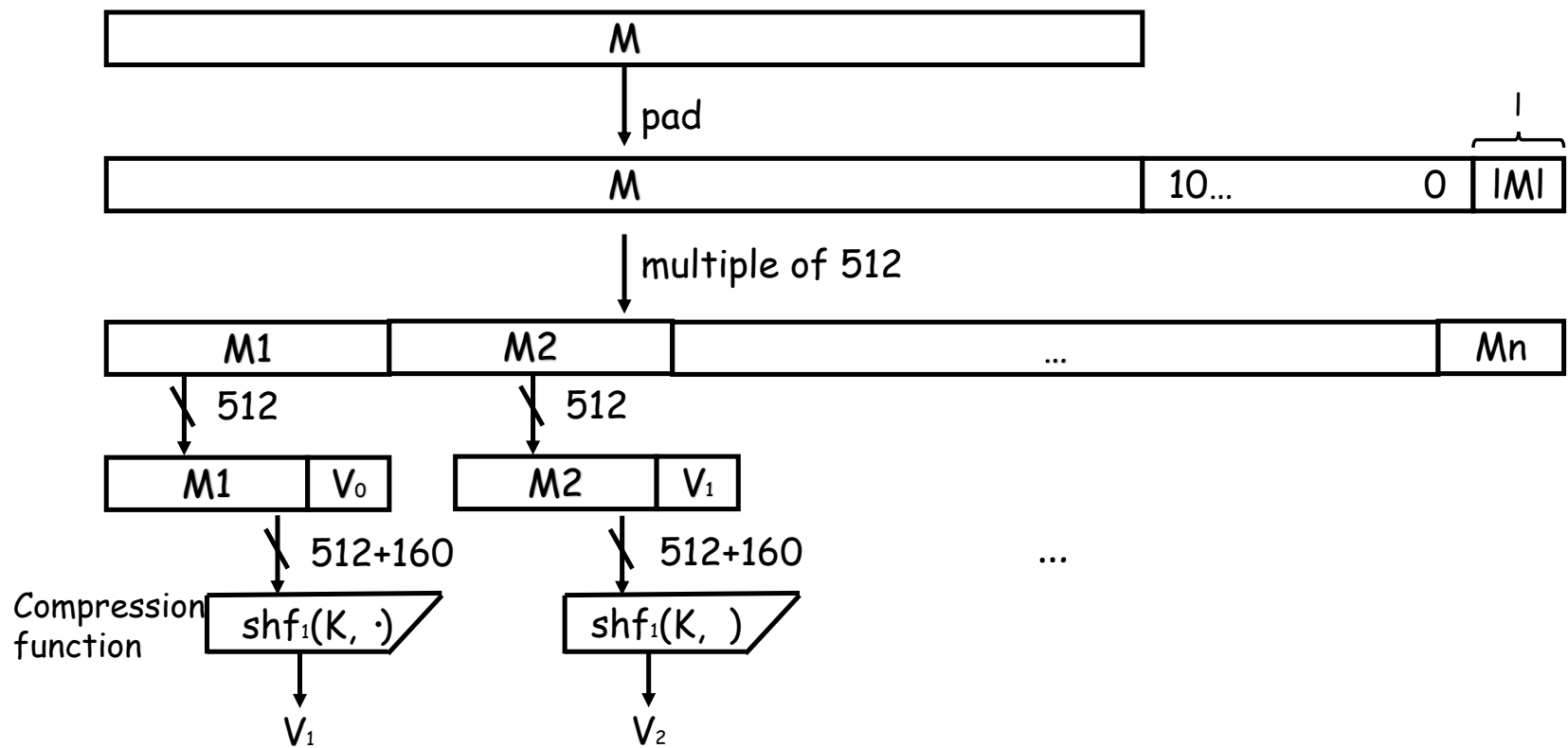


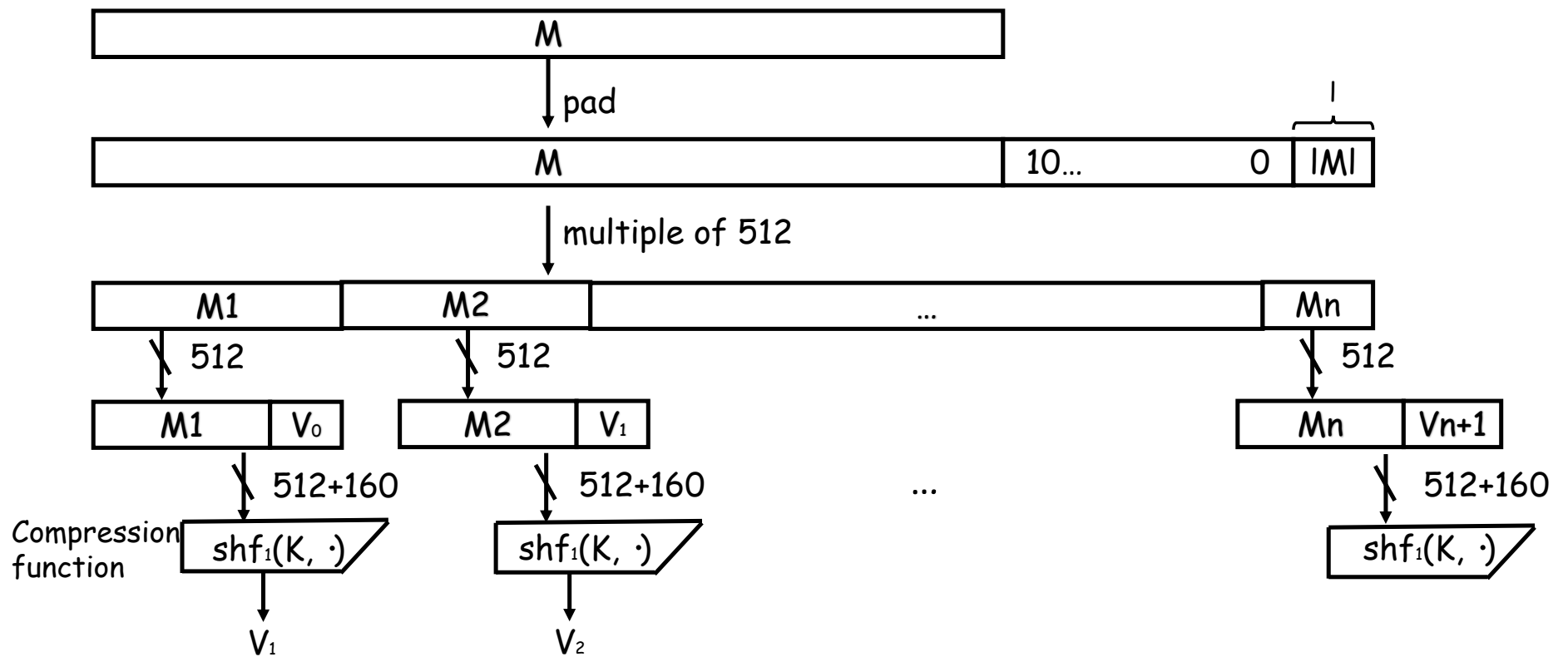
Compression
function

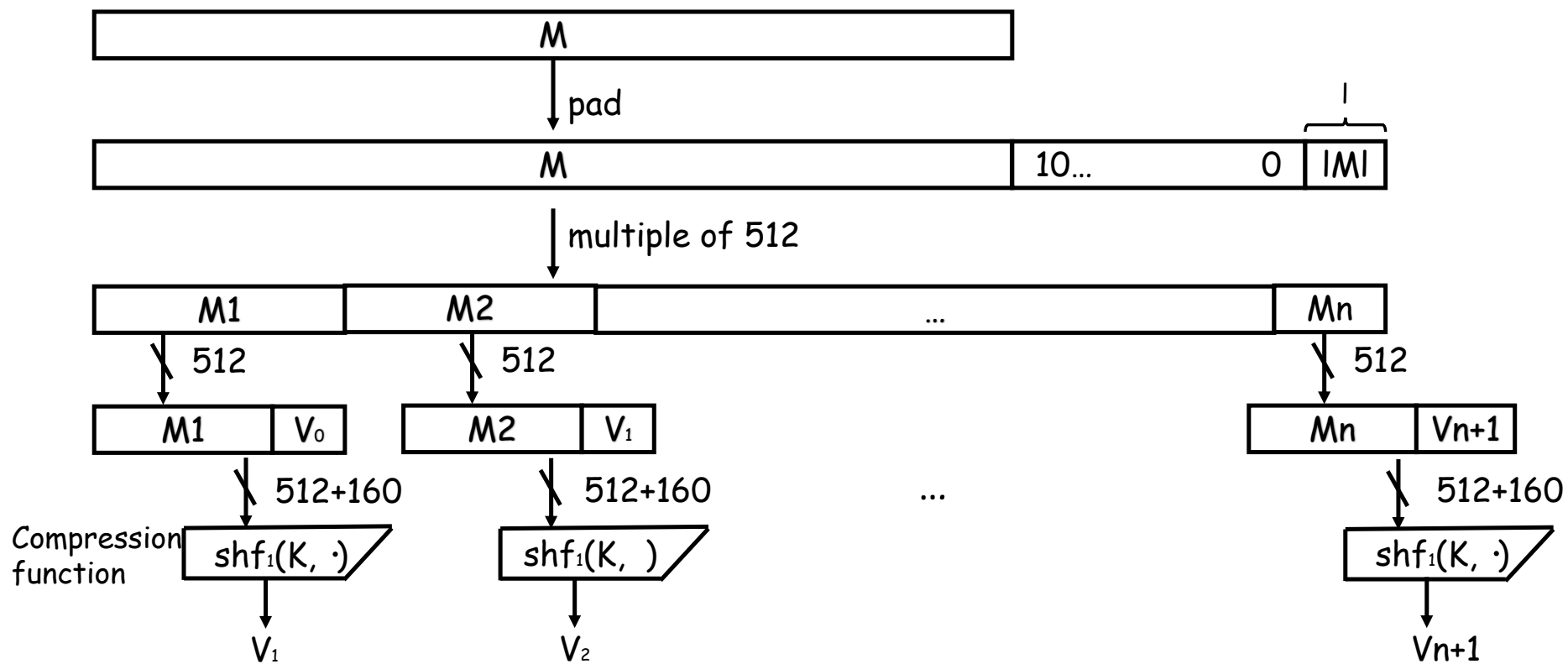


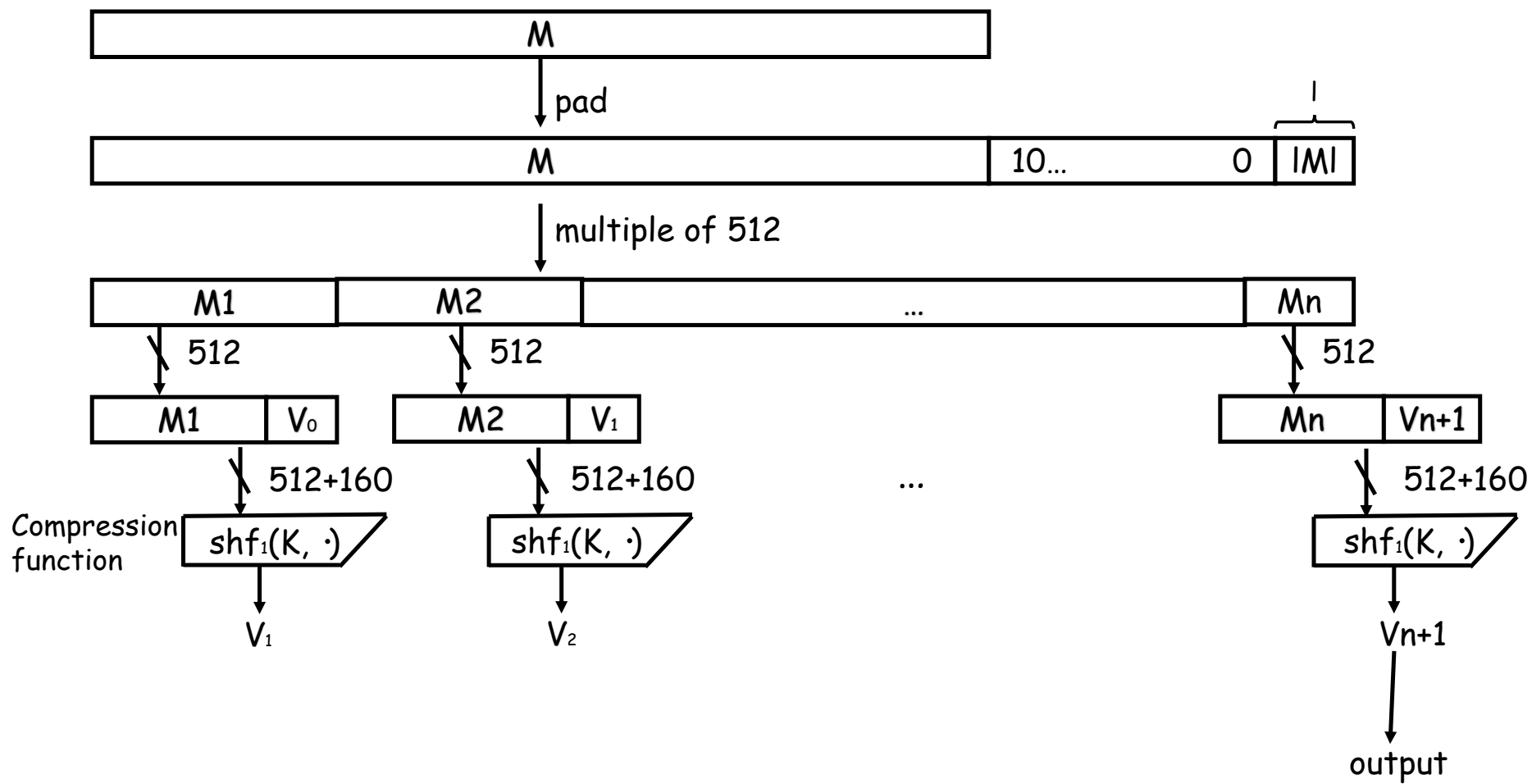


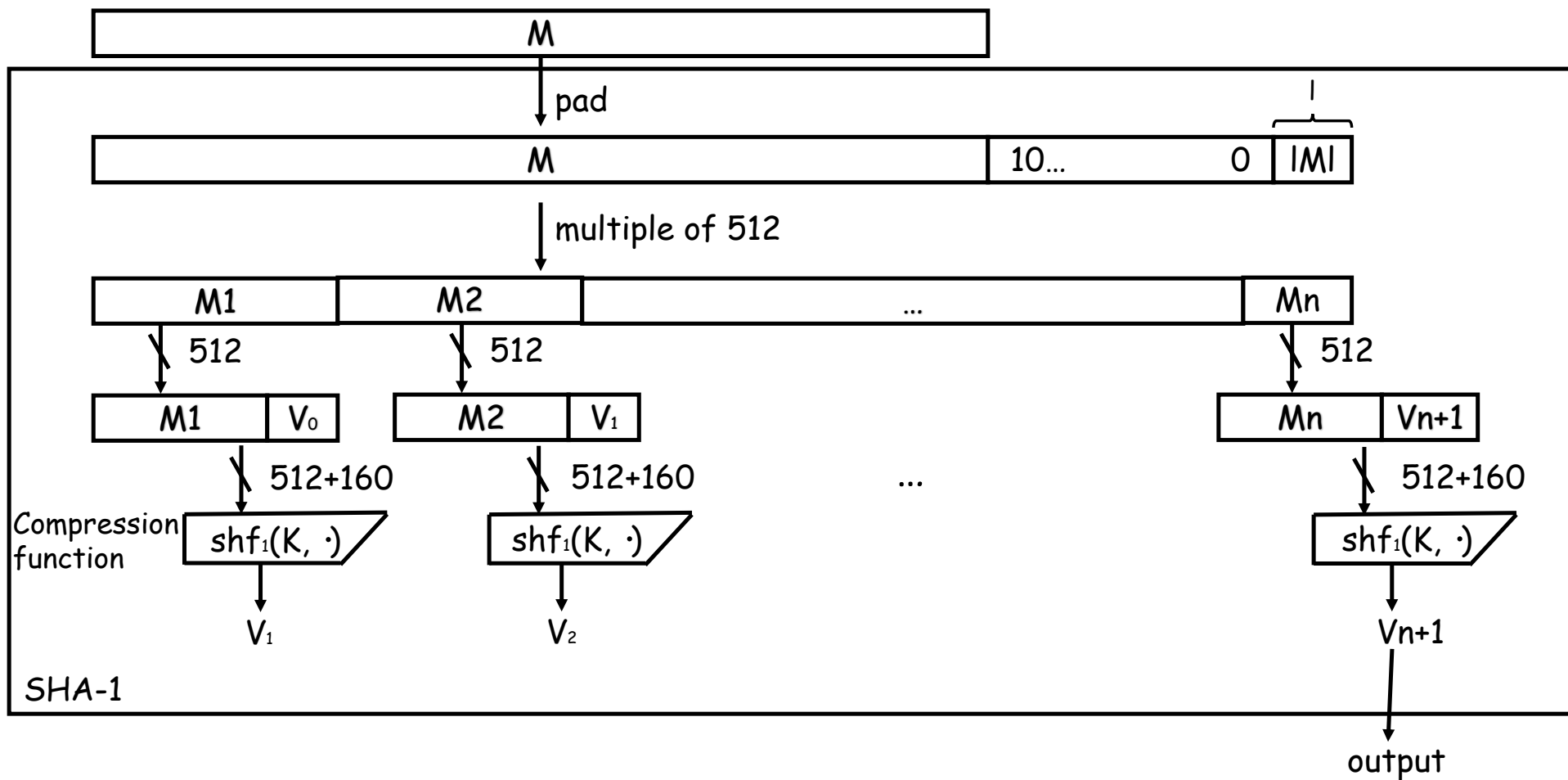












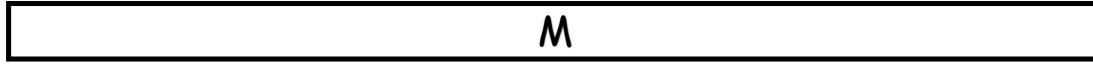
M

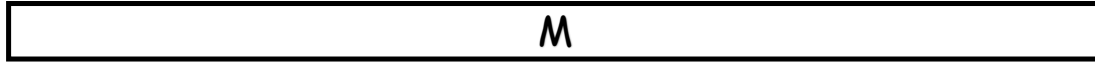


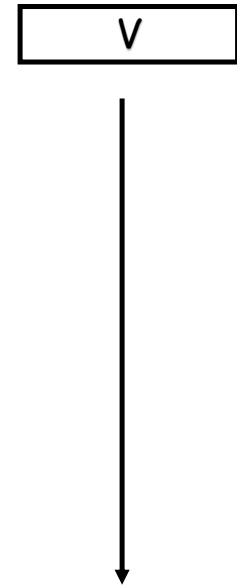
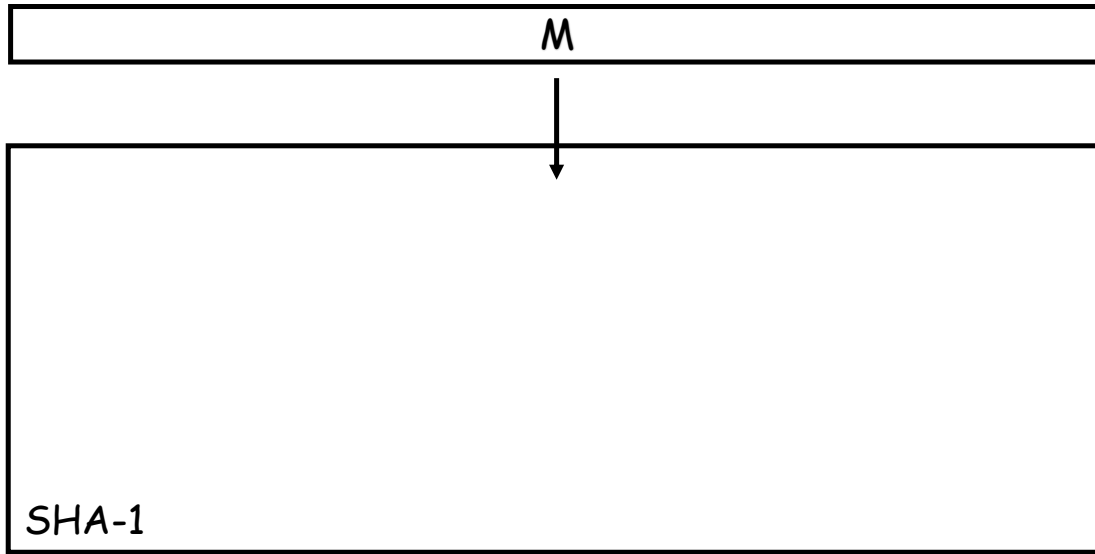
M

V

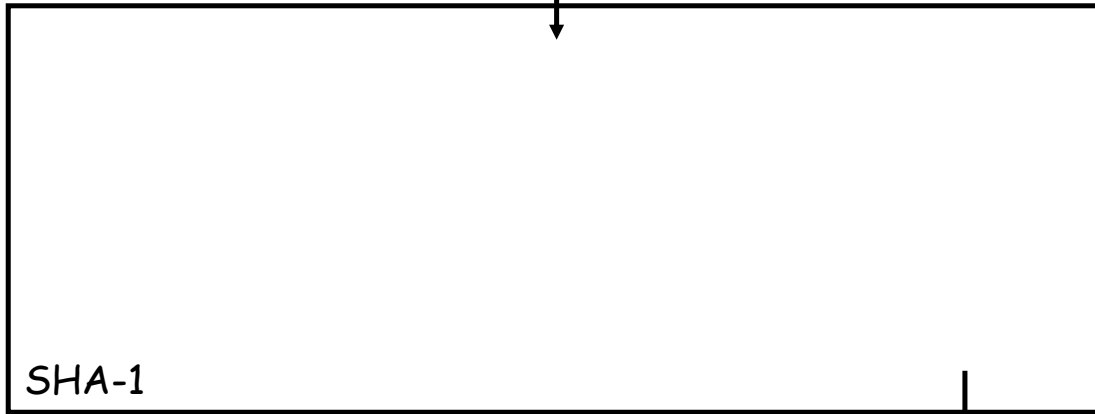








M



SHA-1



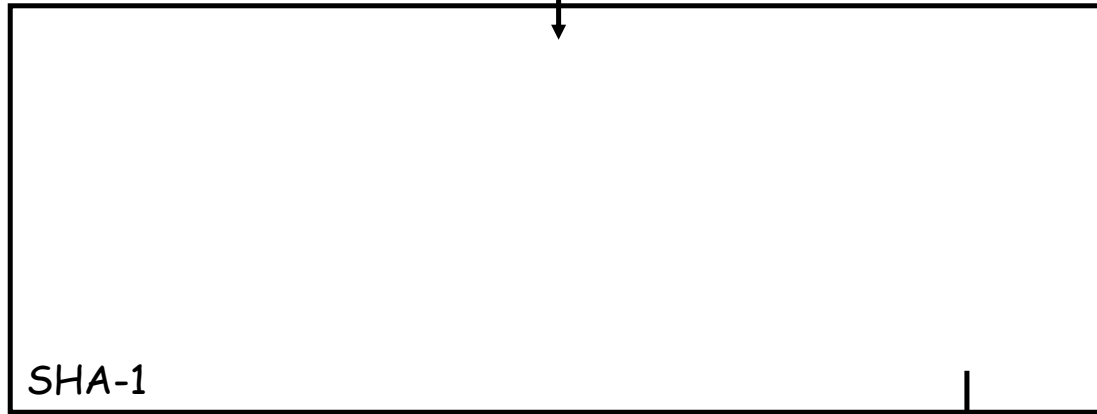
output

V



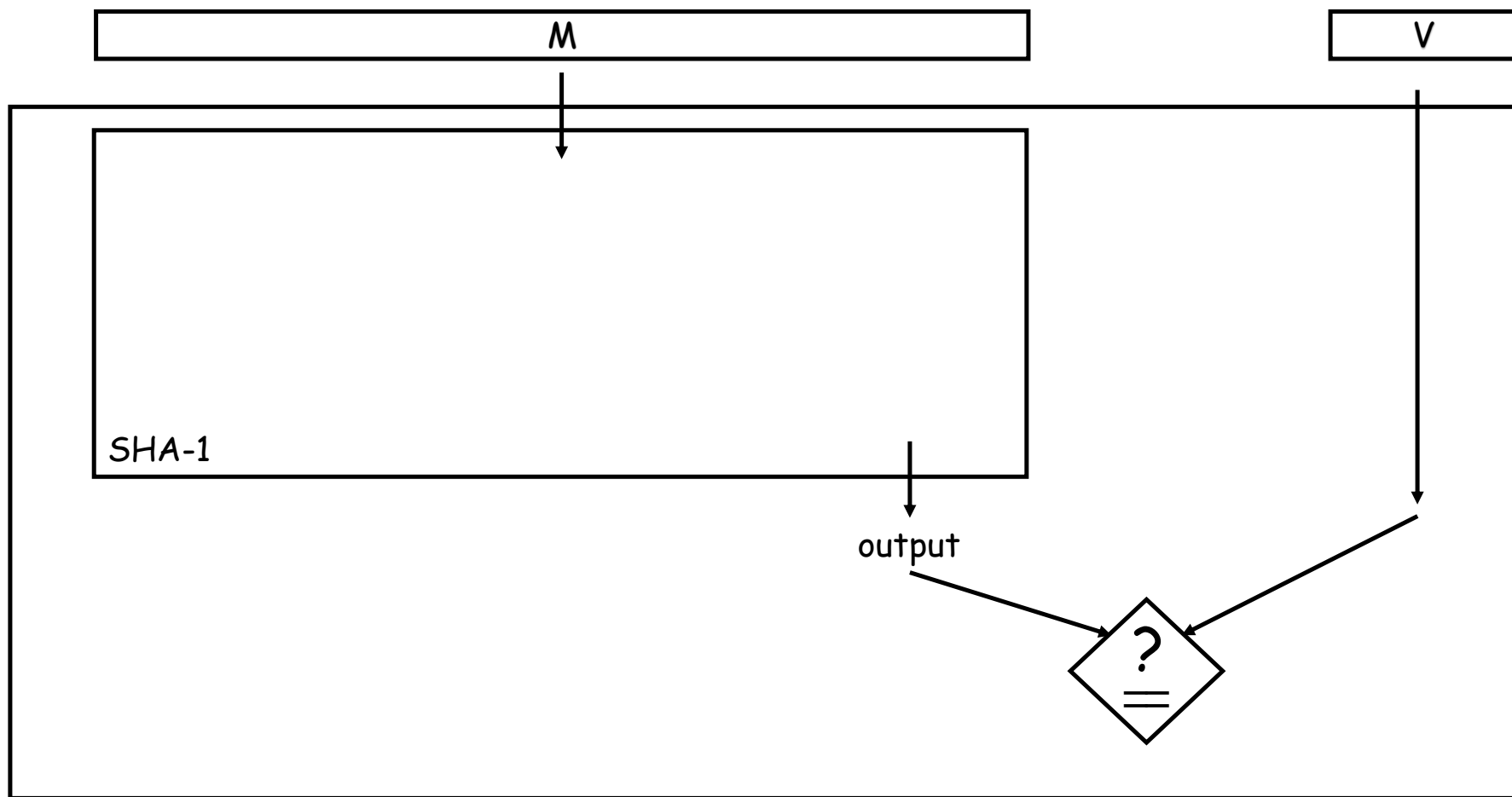
M

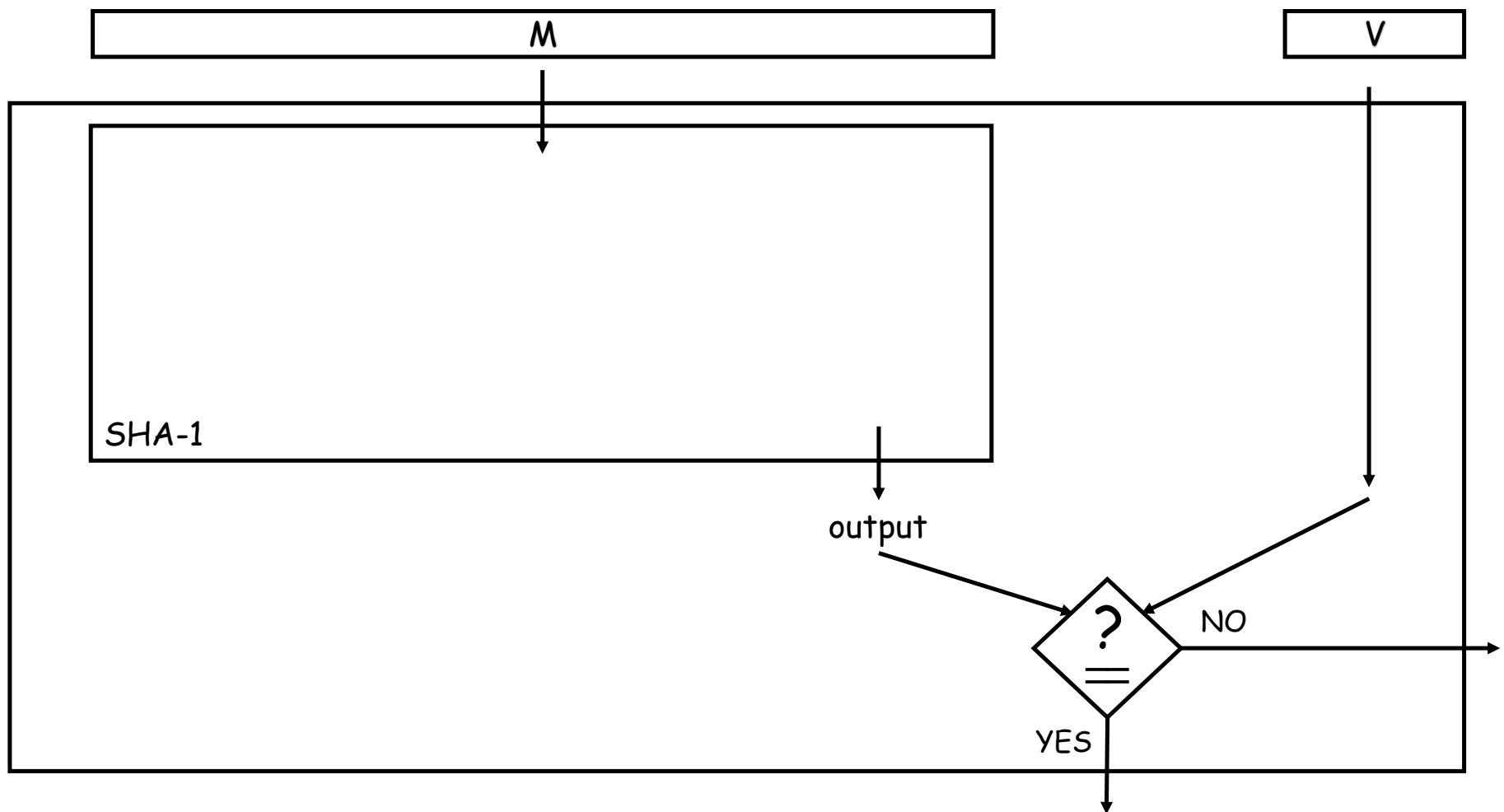
V

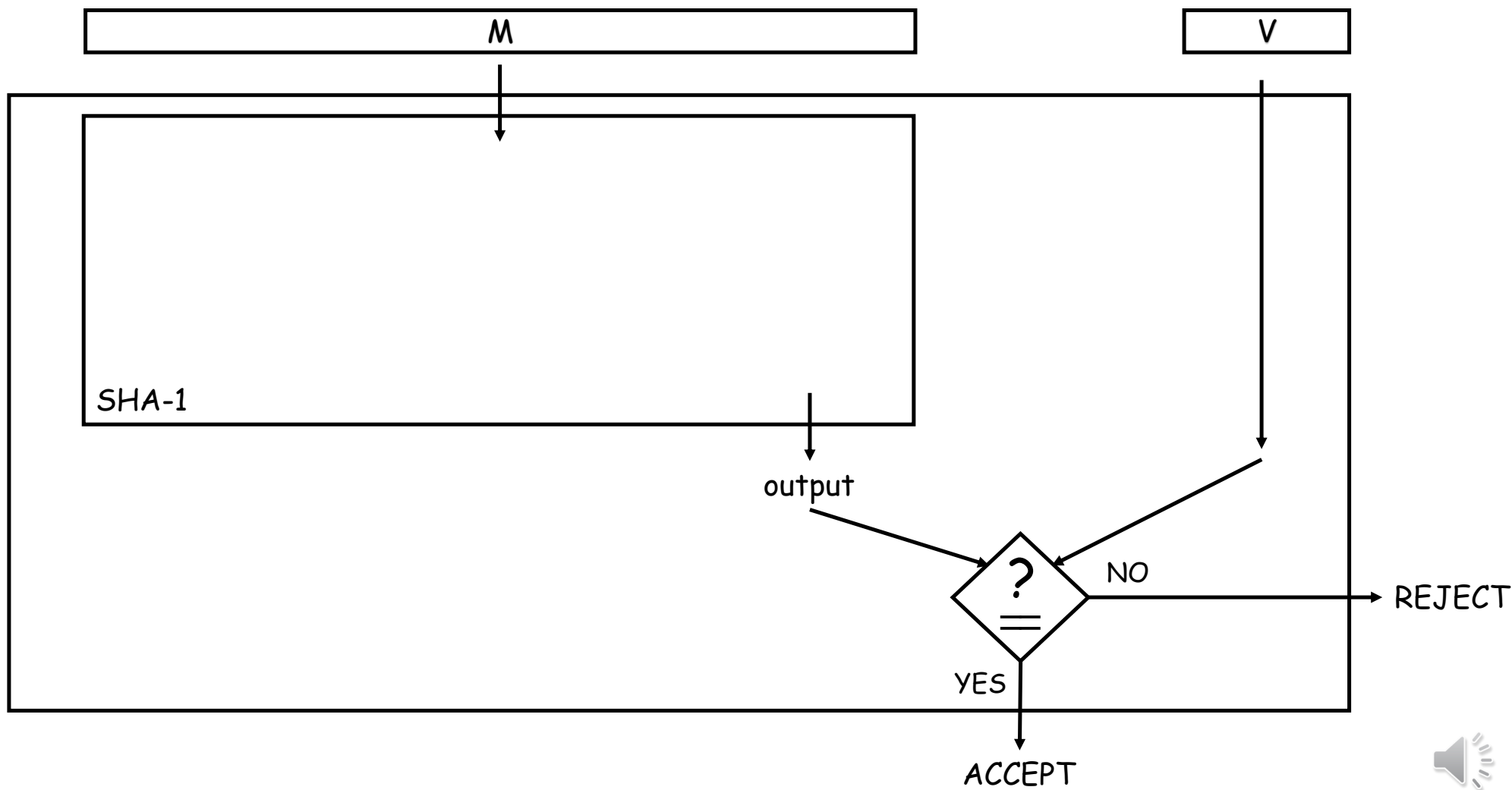


output

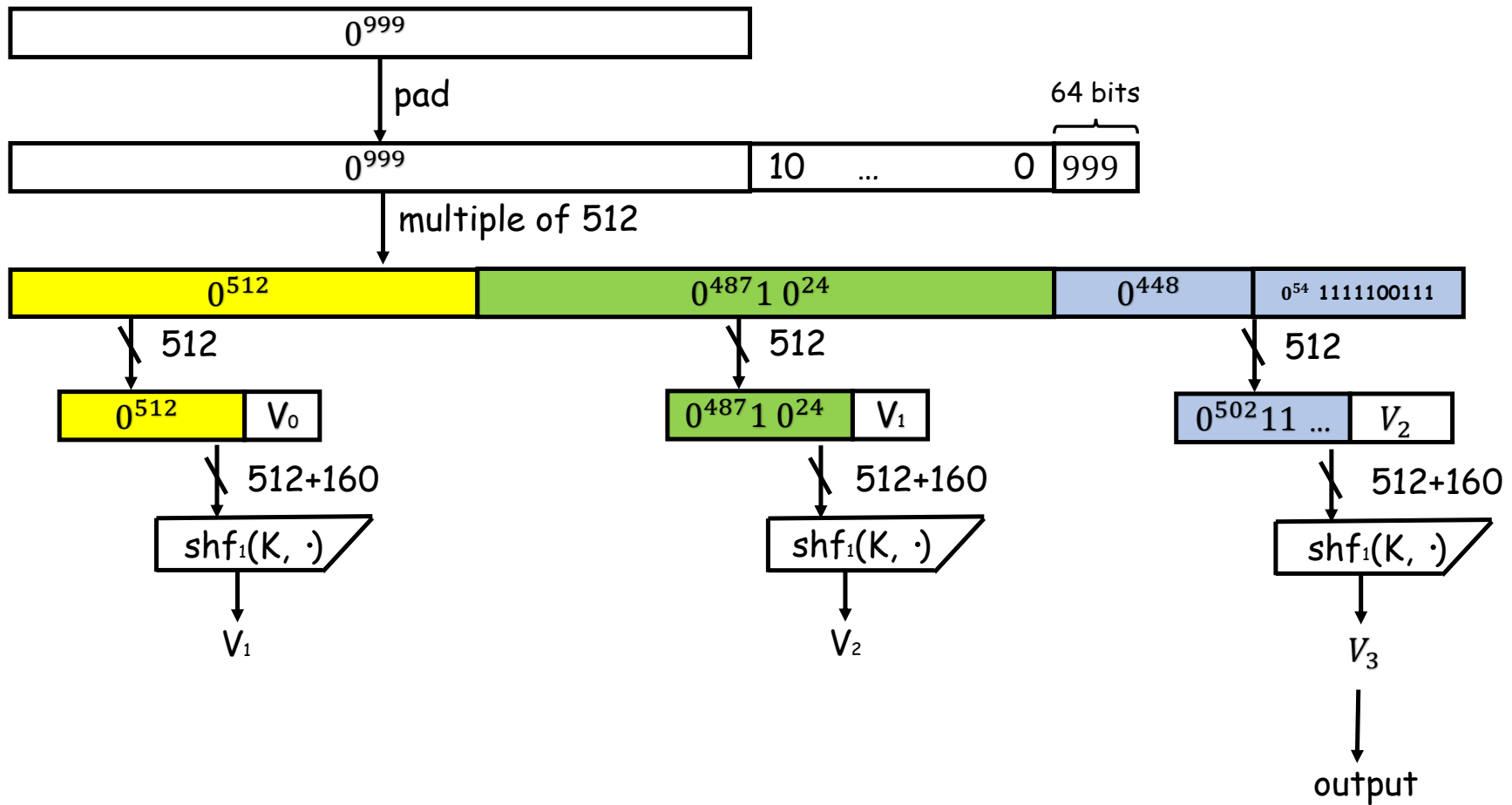








Example: Computing $\text{SHA1}(0^{999}) = V_3$



SHA1

repeatedly apply compression function on each block of message

160 bits $\leftarrow V_0 = 37452301 || \dots || C3D2E1F0$

128 bits $\leftarrow K = 5A827999 || \dots || CA62C1D6$

Observation: use the word “key” but really the “key” is known!

SHA1 in pseudocode

```
algorithm SHA1( $M$ ) //  $|M| < 2^{64}$   
   $V \leftarrow \text{SHF1}(5A827999\|6ED9EBA1\|8F1BBCDC\|CA62C1D6, M)$   
return  $V$ 
```

```
algorithm SHF1( $K, M$ ) //  $|K| = 128$  and  $|M| < 2^{64}$   
   $y \leftarrow \text{shapad}(M)$   
  Parse  $y$  as  $M_1\|M_2\|\dots\|M_n$  where  $|M_i| = 512 (1 \leq i \leq n)$   
   $V \leftarrow 67452301\|EFCDAB89\|98BADCFE\|10325476\|C3D2E1F0$   
  for  $i = 1, \dots, n$  do  
     $V \leftarrow \text{shf1}(K, M_i\|V)$   
return  $V$ 
```

```
algorithm shapad( $M$ ) //  $|M| < 2^{64}$   
   $d \leftarrow (447 - |M|) \bmod 512$   
  Let  $I$  be the 64-bit binary representation of  $|M|$   
   $y \leftarrow M\|1\|0^d\|I$  //  $|y|$  is a multiple of 512  
return  $y$ 
```

The compression function shf1 in SHA1

algorithm shf1($K, B \parallel V$)

// $|K| = 128, |B| = 512, |V| = 160$

Parse B as $W_0 \parallel \dots \parallel W_{15}$ where $|W_i| = 32 (0 \leq i \leq 15)$

Parse V as $V_0 \parallel \dots \parallel V_4$ where $|V_i| = 32 (0 \leq i \leq 4)$

Parse K as $K_0 \parallel \dots \parallel K_3$ where $|K_i| = 32 (0 \leq i \leq 3)$

for $t = 16, \dots, 79$ do

$W_t \leftarrow \text{ROTL}^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$

$A \leftarrow V_0; B \leftarrow V_1; C \leftarrow V_2; D \leftarrow V_3; E \leftarrow V_4$

for $t = 0, \dots, 19$ do

$L_t \leftarrow K_0; L_{t+20} \leftarrow K_1; L_{t+40} \leftarrow K_2; L_{t+60} \leftarrow K_3$

for $t = 0, \dots, 79$ do

if $(0 \leq t \leq 19)$ then $f \leftarrow (B \wedge C) \vee ((\neg B) \wedge D)$

if $(20 \leq t \leq 39)$ OR $(60 \leq t \leq 79)$ then $f \leftarrow B \oplus C \oplus D$

if $(40 \leq t \leq 59)$ then $f \leftarrow (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$

$temp \leftarrow \text{ROTL}^5(A) + f + E + W_t + L_t$

$E \leftarrow D; D \leftarrow C; C \leftarrow \text{ROTL}^{30}(B); B \leftarrow A; A \leftarrow temp$

$V_0 \leftarrow V_0 + A; V_1 \leftarrow V_1 + B; V_2 \leftarrow V_2 + C; V_3 \leftarrow V_3 + D; V_4 \leftarrow V_4 + E$

$V \leftarrow V_0 \parallel V_1 \parallel V_2 \parallel V_3 \parallel V_4$

return V

[All pseudocode is from Bellare-Rogaway lecture notes.]

Real world hash functions

Hash function	Number of output bits
MD4	128
MD5	128
SHA1	160
SHA256	256
SHA512	512
SHA3-256	256
SHA3-512	512

Note some terminology:

- ▶ SHA2 family members = SHA256, SHA512
- ▶ SHA3 family members = SHA3-256, SHA3-512

Security definitions for hash functions

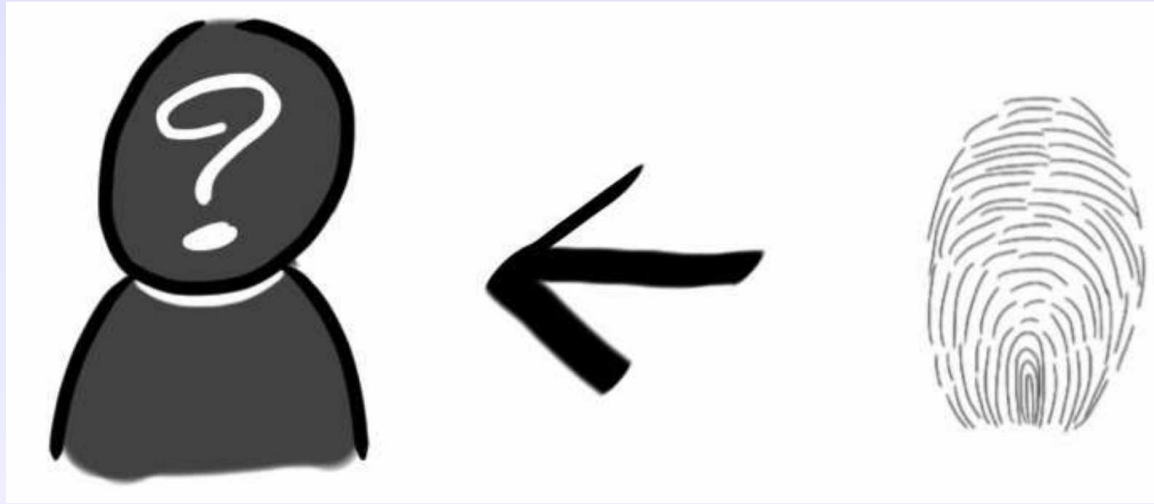
Security definitions for hash functions

What do we expect from hash functions?

Recall the common applications:

- ▶ password storage
- ▶ allowing people to check integrity of software that they download

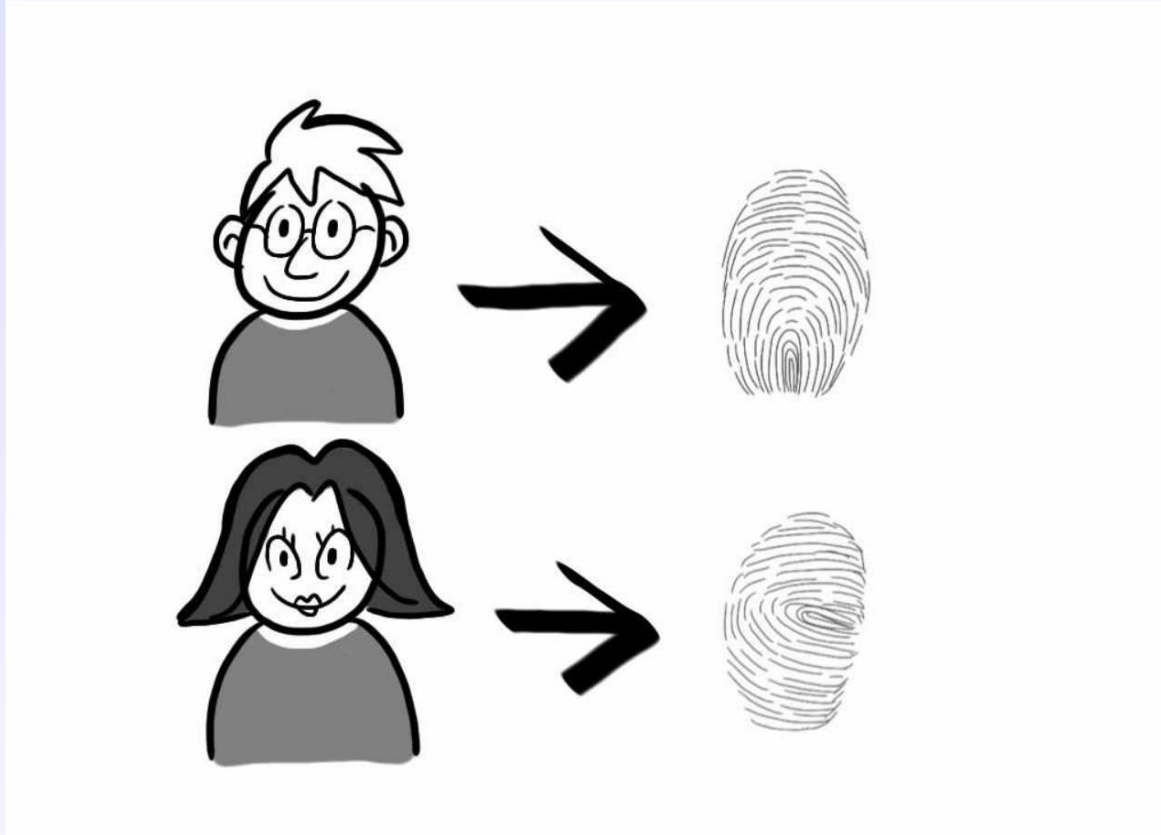
Security definition: Pre-image resistance



Preimage resistance means that it should be difficult to figure out what was hashed simply by looking at the hash value.

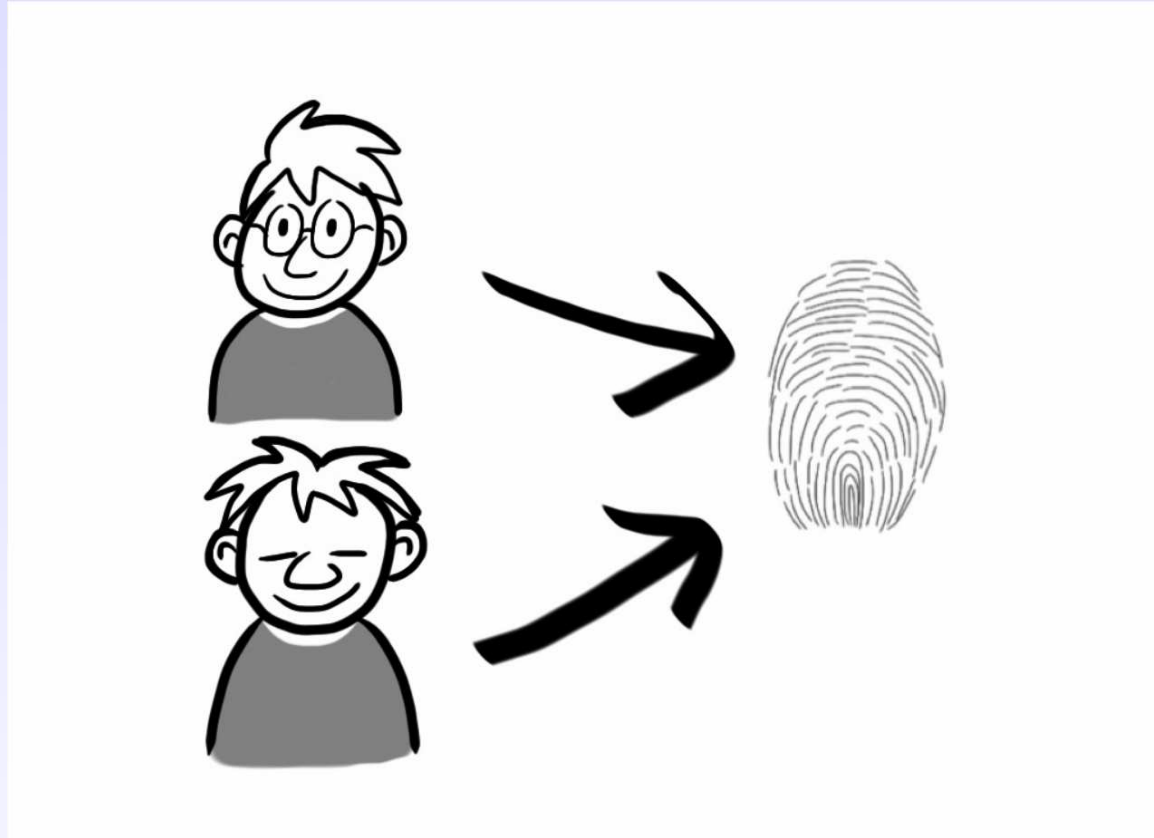
Example application: password storage

Security definition: collision resistance



For collision resistant hash functions, one would expect that, in most cases, hashing two distinct input values would result in two different hash values.

Security definition: collision resistance (cont.)



A collision can always occur, however, since a hash function maps values from a large set to those in a smaller set.

Example application: software integrity check

Example: SHA1

Definition

A **collision** for a function $h : \mathcal{D} \rightarrow \mathcal{R} \equiv$ a pair $x_1, x_2 \in \mathcal{D}$ such that $h(x_1) = h(x_2)$ but $x_1 \neq x_2$

Collision resistance of SHA1:

It is hard to find M and M' such that $\text{SHA1}(M) = \text{SHA1}(M')$ but $M \neq M'$

There are **many** such M and M' ! (by pigeonhole principle).

Security: resistance against pre-image attacks

Let m, n be integers such that $m > n$.

Let $H : \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a hash function.

Subroutine *Initialize*

$x \xleftarrow{\$} \{0, 1\}^m ; h \leftarrow H(x)$
Return h

Subroutine *Finalize*(x')

Return $(H(x') = h)$

Experiment $\mathbf{Exp}_H^{\text{pre}}(A)$

$h \leftarrow \text{Initialize}$

$x' \xleftarrow{\$} A(h)$

Return *Finalize*(x')

pre-image advantage

The pre-image advantage of an adversary A mounting a pre-image attack against H is

$$\mathbf{Adv}_H^{\text{pre}}(A) = \Pr [\mathbf{Exp}_H^{\text{pre}}(A) \Rightarrow \text{true}] .$$

Security: resistance against second pre-image attacks

Let m, n be integers such that $m > n$.

Let $H : \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a hash function.

Subroutine *Initialize*

$x \xleftarrow{\$} \{0, 1\}^m$

Return x

Subroutine *Finalize*(x')

Return $(H(x) = H(x') \wedge x \neq x')$

Experiment $\mathbf{Exp}_H^{\text{sec}}(A)$

$x \leftarrow \text{Initialize}$

$x' \xleftarrow{\$} A(x)$

Return *Finalize*(x')

second pre-image advantage

The second pre-image advantage of an adversary A mounting a second pre-image attack against H is

$$\mathbf{Adv}_H^{\text{sec}}(A) = \Pr [\mathbf{Exp}_H^{\text{sec}}(A) \Rightarrow \text{true}] .$$

Security: collision resistance

Let m, n be integers such that $m > n$.

Let $H : \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a hash function.

Subroutine *Initialize*
Return

Subroutine *Finalize*(x, x')
Return $(H(x) = H(x') \wedge x \neq x')$

Experiment $\mathbf{Exp}_H^{\text{coll}}(A)$

Initialize

$(x, x') \xleftarrow{\$} A$

Return *Finalize*(x, x')

collision advantage

The collision advantage of an adversary A mounting a collision attack against H is

$$\mathbf{Adv}_H^{\text{coll}}(A) = \Pr \left[\mathbf{Exp}_H^{\text{coll}}(A) \Rightarrow \text{true} \right] .$$

Toy example

Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a block cipher.

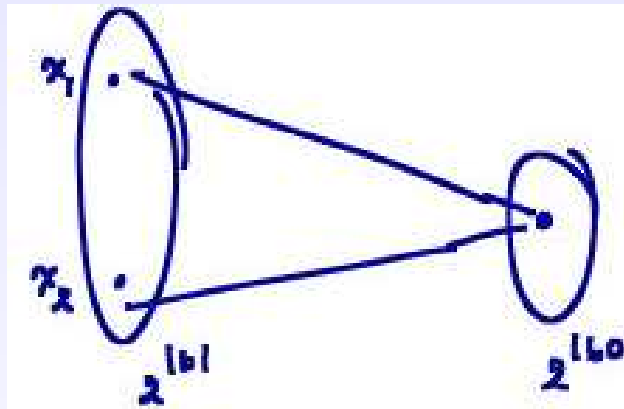
Let $K \in \{0, 1\}^k$.

Define a hash function $H : \{0, 1\}^{mn} \rightarrow \{0, 1\}^n$ based on K and E for any positive integer m using the raw CBC MAC construction.

Is H collision resistant?

Collision-finding attacks: Example

Example: Suppose $|\mathcal{D}| = 2|\mathcal{R}|$



Collision-finding attack: Example (cont.) – strategy #1

Strategy # 1

Pick a point, go through all elements in the domain in some order until collide.

Worst case :

- ▶ the last one is the one and
- ▶ **regular**, i.e., evenly distributed.

[In this case, only 2 points collide for each element y in \mathcal{R} ,
i.e., $|H_K^{-1}(y)| = 2$]

number of trials needed $= 2^{161}$

Collision-finding attack: Example (cont.) – strategy #2

Strategy # 2

Pick a point, pick another point at random from \mathcal{D} until collide.

Find collision with probability about 1 in $|\mathcal{R}|$. So,

$$\text{number of trials needed} = 2^{160}$$

Collision-finding attack: Example (cont.) – strategy #3

Strategy # 3: Birthday attack

Pick random points from \mathcal{D} until find a pair that collides.

number of trials needed = $O(\sqrt{|\mathcal{R}|}) = O(2^{80}) < 2^{160}$

Birthday paradox

Let $x_1, \dots, x_q \in \mathcal{D}$ be independent identically distributed elements of \mathcal{D} .

Birthday bound

If $q = 1.2 \times |\mathcal{D}|^{1/2}$, then

$$\Pr [\exists i \neq j : x_i = x_j] \geq \frac{1}{2} .$$

Proof of birthday bound (for uniform independent x 's)

Denote $|\mathcal{D}|$ by D . Let E_i be the event that there is no collision after having picked i values. Let q be the total number of values that we pick.

$$\begin{aligned}\Pr[\exists i \neq j : x_i = x_j] &= 1 - \Pr[\forall i \neq j : x_i \neq x_j] \\&= 1 - \Pr[E_q] \\&= 1 - \Pr[E_q \mid E_{q-1}] \cdot \Pr[E_{q-1}] \\&= 1 - \left(\frac{D-1}{D}\right) \left(\frac{D-2}{D}\right) \cdots \left(\frac{D-n+1}{D}\right) \\&= 1 - \prod_{i=1}^{q-1} \left(1 - \frac{i}{D}\right) \\&\geq 1 - \prod_{i=1}^{q-1} e^{-\frac{i}{D}} \quad [\text{because } 1 - x \leq e^{-x}] \\&= 1 - e^{-\frac{1}{D} \sum_{i=1}^{q-1} i} \geq 1 - e^{-\frac{q^2}{2D}} \geq 1 - e^{-0.72} = 0.53\end{aligned}$$

So we get this bound when $q^2/2D = 0.72$, i.e., $q = 1.2 \times |\mathcal{D}|^{1/2}$.

Birthday attack against hash functions

Let $H : \mathcal{D} \rightarrow \{0, 1\}^n$ be a hash function.

Birthday attack

Adversary A

For $i = 1, \dots, q$ do $x_i \xleftarrow{\$} \mathcal{D}$; $y_i \leftarrow H(x_i)$

If $\exists i, j$ such that $i \neq j$ and $x_i \neq x_j$ but $y_i = y_j$
then return x_i, x_j

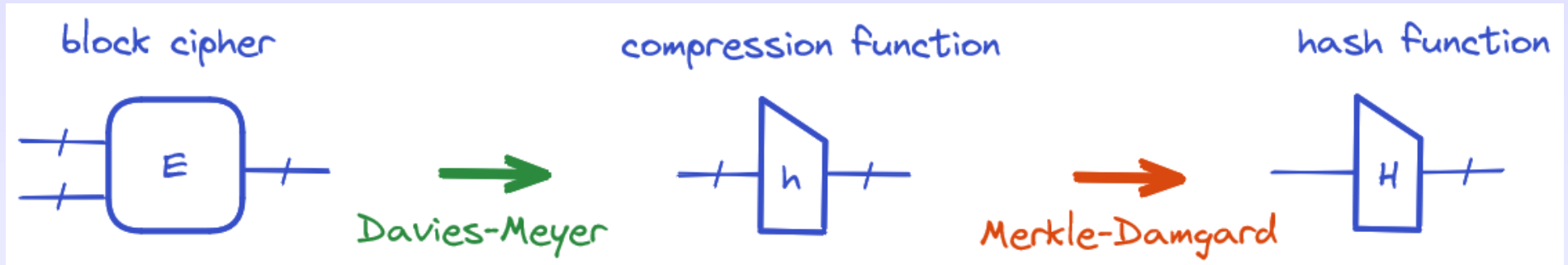
return \perp

Assuming that H is regular, according to the birthday bound, A should find a collision in about $q = 1.2 \times \sqrt{|\mathcal{D}|}$ tries.

Birthday attack against **hash functions** is analogous to **exhaustive key search attack** against **block ciphers**.

Building hash functions

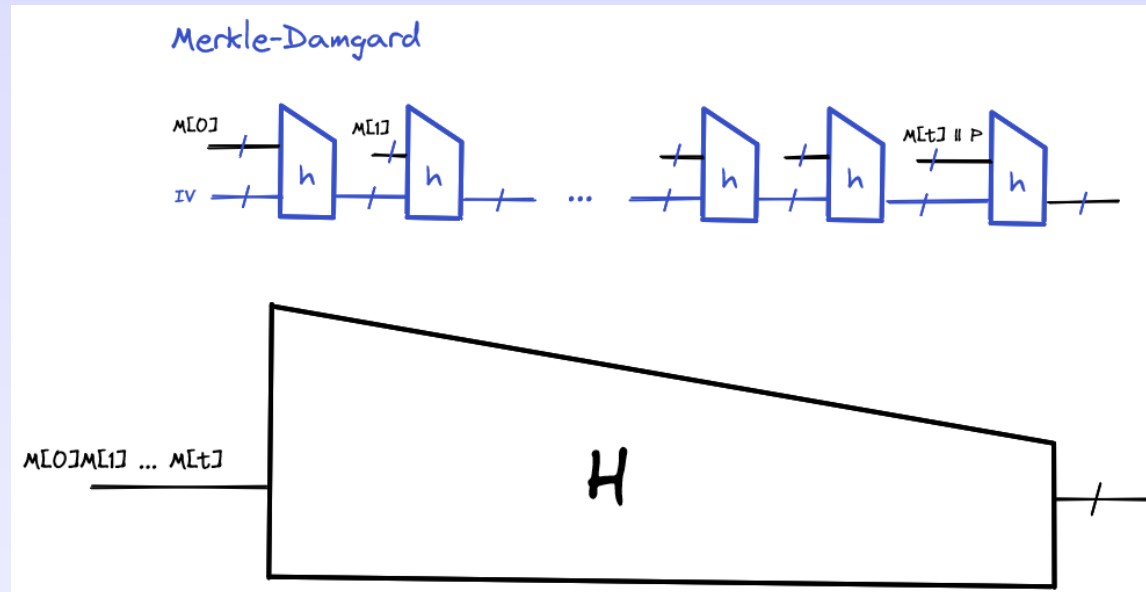
A common paradigm for building hash functions



What has been proven

- ▶ **Merkle-Damgard**: If h is collision-resistant, then so is H .
- ▶ **Davies-Meyer**: In the ideal cipher model (i.e., E is a public random permutation), h is collision-resistant.

Compression function \Rightarrow hash function: MD transform



Theorem

Let $h : \{0, 1\}^k \times \{0, 1\}^{b+n} \rightarrow \{0, 1\}^n$ be a family of functions, and let $H : \{0, 1\}^k \times \mathcal{D} \rightarrow \{0, 1\}^n$ be the result of applying the MD transform to h . Given a CR adversary A against H , then we can construct a CR adversary B against h such that

$$\mathbf{Adv}_H^{\text{coll}}(A) \leq \mathbf{Adv}_h^{\text{coll}}(B)$$

and the running time of B is the running time of A plus the time for computing h on the outputs of A .

Block cipher \Rightarrow compression function

Consider SHA256, our compression function must compress $512+256$ bits to 256 bits.

Example

Let $E : \{0, 1\}^{512} \times \{0, 1\}^{256} \rightarrow \{0, 1\}^{256}$ be a block cipher. Let $h : \{0, 1\}^{512+256} \rightarrow \{0, 1\}^{256}$ be defined as follows:

$$h(x||v) = E_x(v) .$$

Recall the security definition for collision resistant against a compression function.

Would this construction be collision resistant?

Block cipher \Rightarrow compression function: Davies-Meyer

Consider SHA256, our compression function must compress 512+256 bits to 256 bits.



Davies-Meyer construction

Let $E : \{0, 1\}^{512} \times \{0, 1\}^{256} \rightarrow \{0, 1\}^{256}$ be a block cipher. Let $h : \{0, 1\}^{512+256} \rightarrow \{0, 1\}^{256}$ be defined as follows:

$$h(x||v) = E_x(v) \oplus v$$

where \oplus above is actually “component-wise addition modulo 2^{32} .”

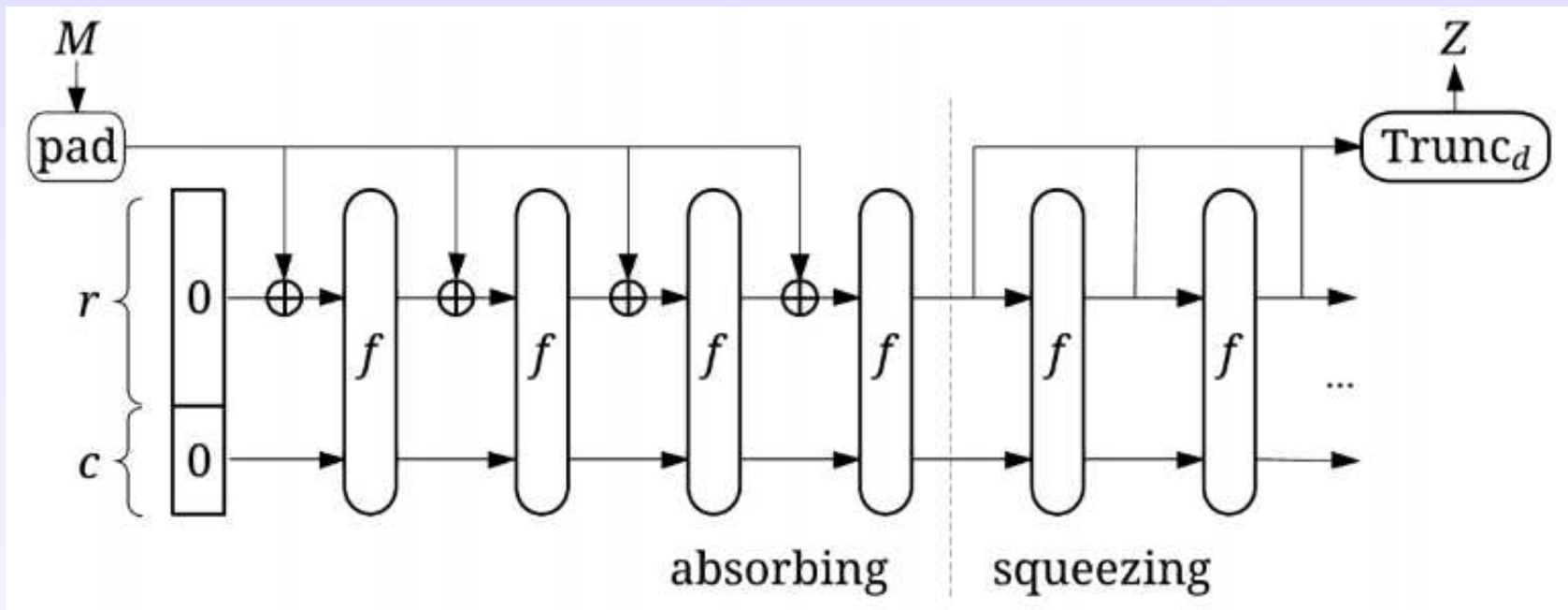
Security proof for Davies-Meyer construction?

Q: Can we prove that Davies-Meyer construction yields a collision resistant compression function assuming that E is a PRF?

A: No! PRF does not cut it. We can construct a PRF secure E such that h is not collision resistant.

SHA3

SHA3 does not follow this paradigm.



- ▶ $f : \{0, 1\}^{r+c} \rightarrow \{0, 1\}^{r+c}$ is a public and invertible permutation.
- ▶ d is the number of output bits.
- ▶ $c = 2d$