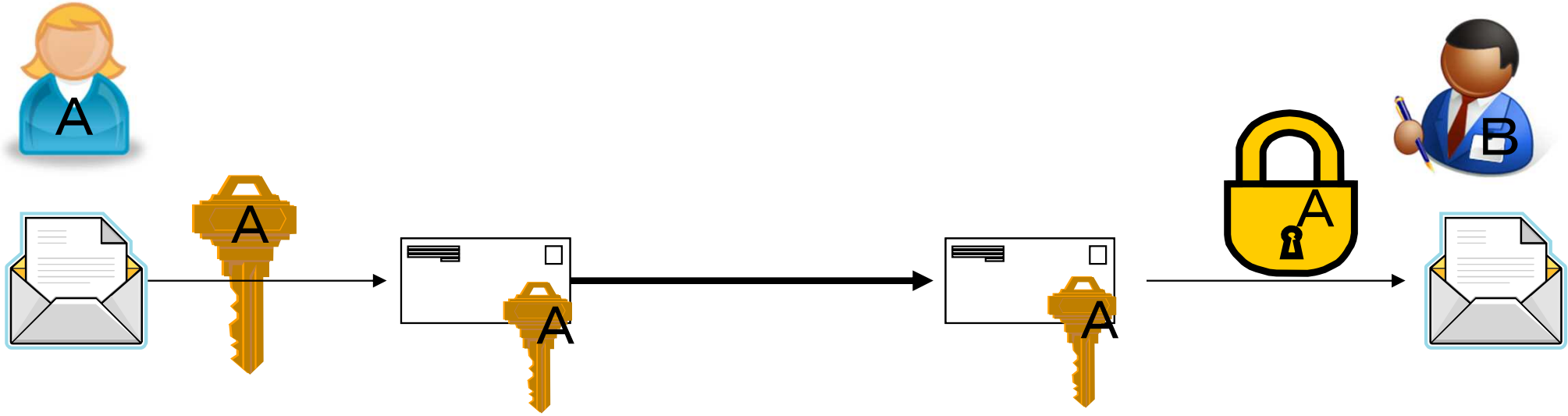
ディジタル署名とハッシュ関数

公開鍵暗号による電子署名



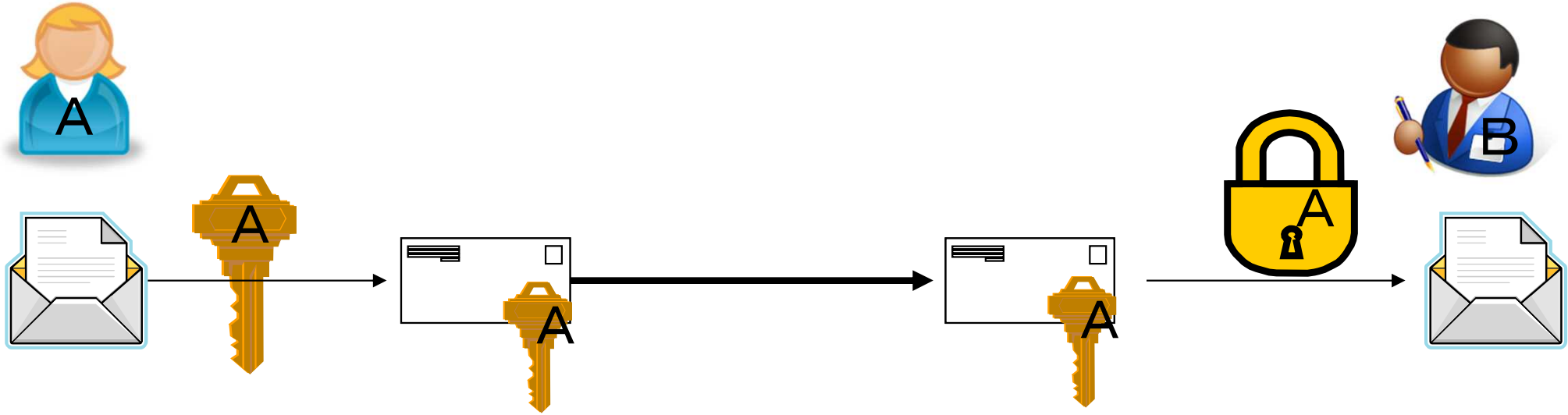
自分の秘密鍵で“変換”して　　相手の公開鍵で復号

Bが正しく復号できれば、
Aが（Aしか持っていない）秘密鍵で“変換”したということ

↓

受信した文書は、たしかにAが送ったものである

RSA暗号による電子署名（1）

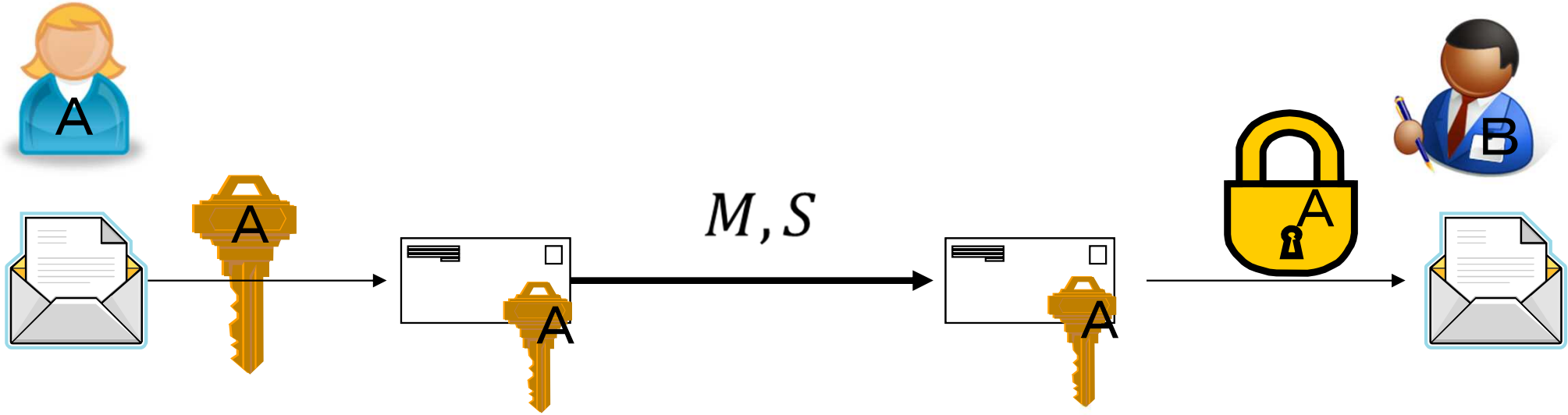


$$M \qquad S = M^d \pmod{n} \qquad S^e = \left(M^d\right)^e = M \pmod{n}$$

Bobは、Aliceの署名の確認はできるが、Aliceの署名を作成することはできない

署名の確認時に、別途Mが必要になるので効率が悪い　→ハッシュ関数の利用

RSA暗号による電子署名（2）



M

$S = h(M)^d \ (mod\ n)$

$S^e = (h(M)^d)^e$
$= h(M)\ (mod\ n)$

$S^e (mod\ n) = h(M)$
をチェックする

ElGamal署名

準備（ElGamal暗号と同様）

p ：大きな素数　　g ：法pにおける原始元

a ：秘密鍵 $(1 \le a < p-1)$　　$y = g^a \pmod p$

$h(\cdot)$ ：安全なハッシュ関数（値域：*1～p-1*）

署名

M ：文書　　k ：乱数 $(1 \le k < p-1, (k, p-1) = 1)$

$$r = g^k \pmod p$$

$$s = k^{-1}(h(M) - a \cdot r) \pmod{p-1}$$

(r, s) ：署名

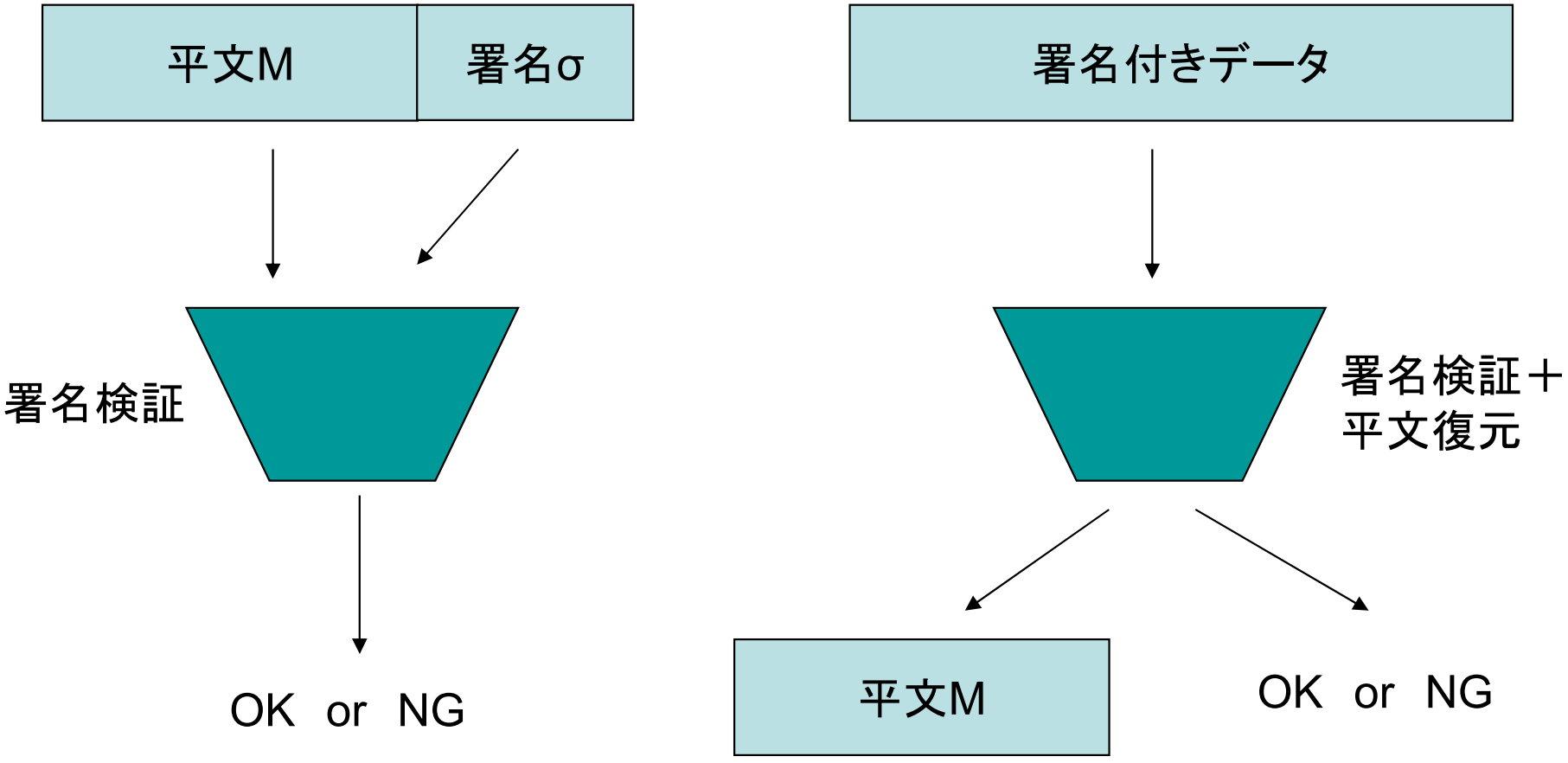
検証

$$y^r \cdot r^s \stackrel{?}{=} g^{h(M)} \pmod p$$

（左辺）

$$y^r \cdot r^s = g^{a \cdot r} \cdot g^{k \cdot k^{-1}(h(M) - a \cdot r)} = g^{h(M)} \pmod p$$

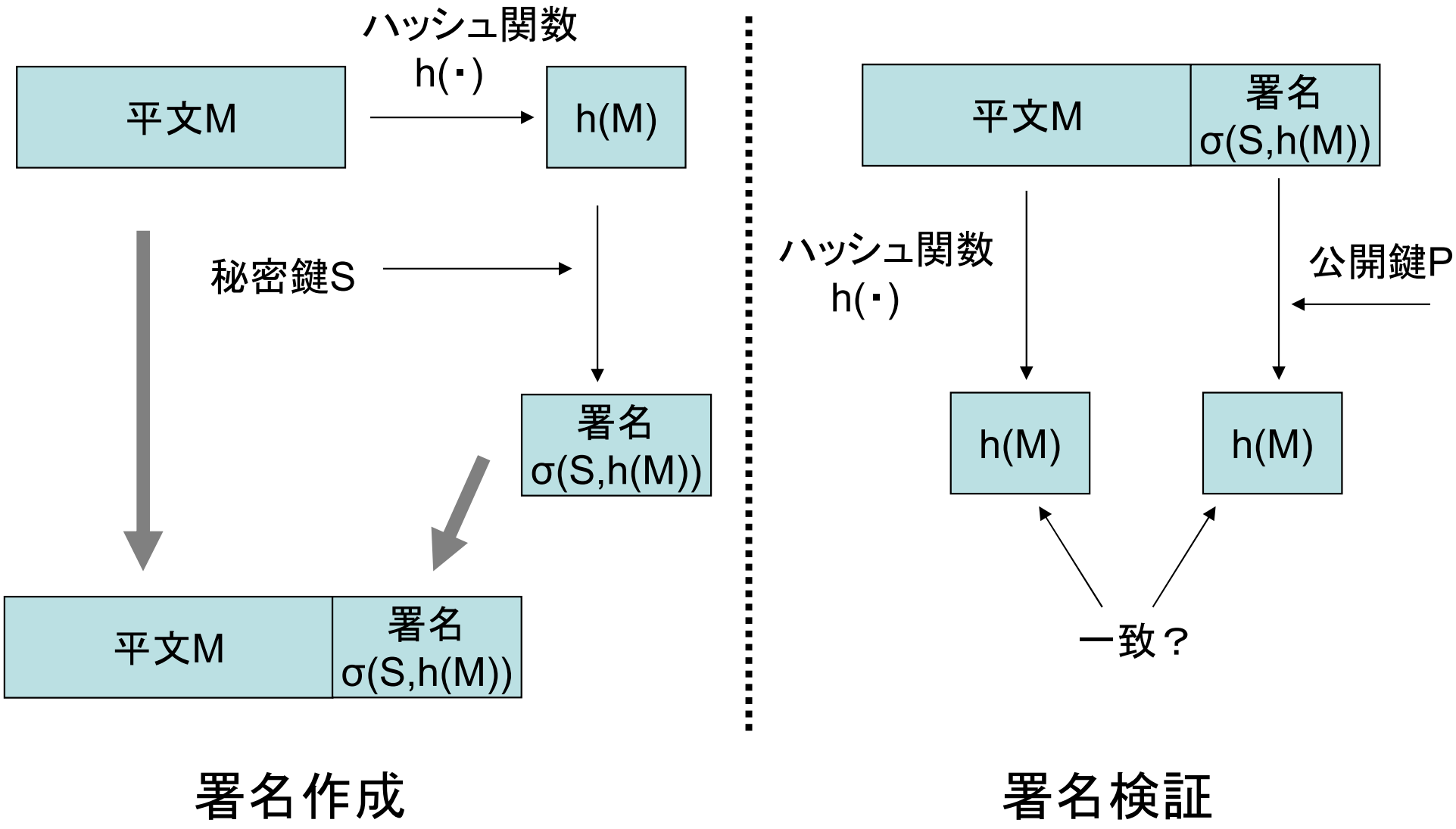
ディジタル署名の種類



付録型署名

メッセージ回復型署名

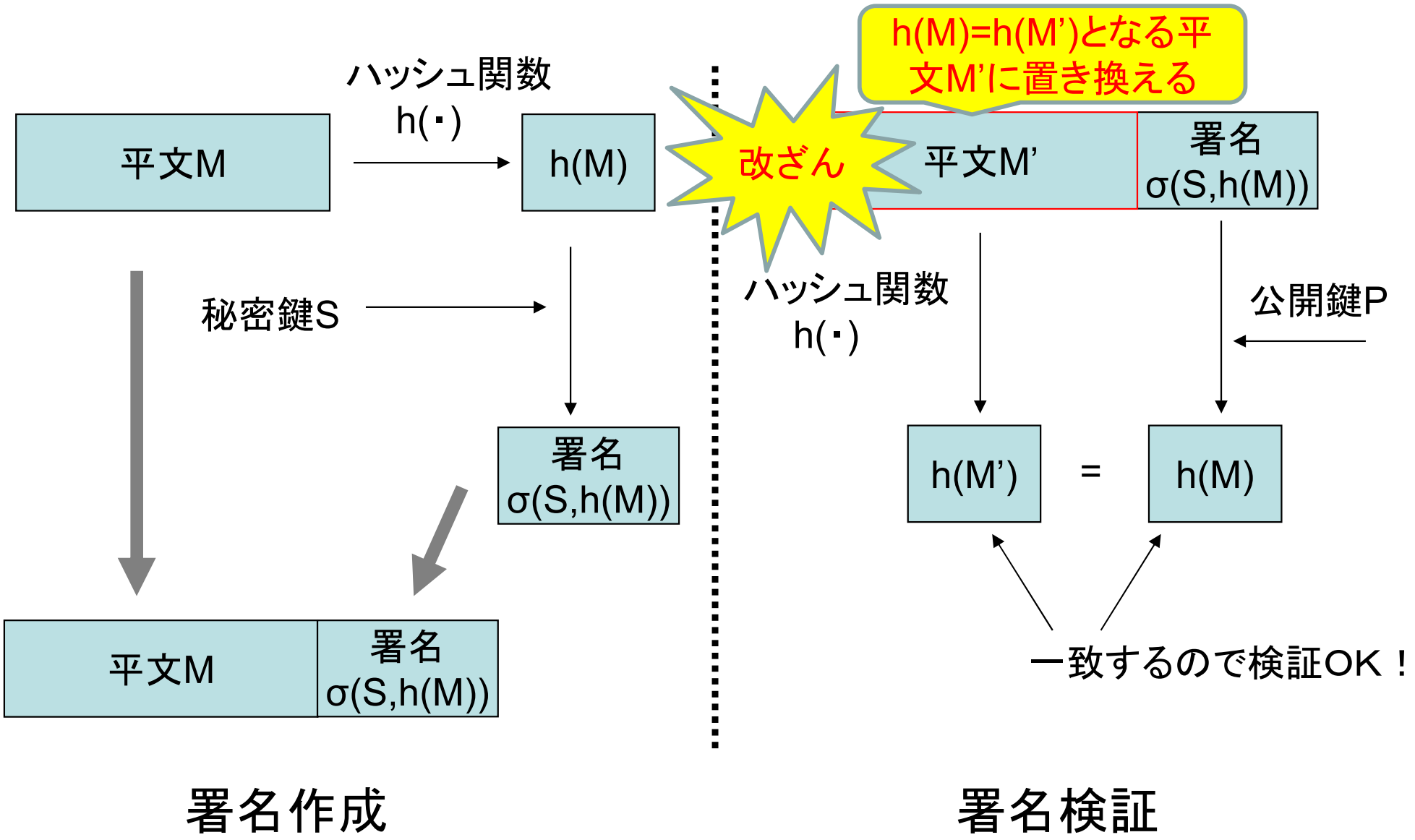
ディジタル署名の流れ


ハッシュ関数
h(・)
平文M
h(M)
秘密鍵S
署名
σ(S,h(M))
平文M
署名
σ(S,h(M))
署名作成
平文M
署名
σ(S,h(M))
ハッシュ関数
h(・)
公開鍵P
h(M)
h(M)
一致？
署名検証

ハッシュ関数とは

- メッセージを小さなサイズ（＝ハッシュサイズ）に要約する関数。要約された値はハッシュ値
- ハッシュ関数の安全性
  - 不可逆性：与えられたハッシュ値H=h(x)からh(x')=Hを満たすx'を見つけるのが難しい
  - 二次不可逆性：与えられたxからh(x)=h(x')となるx'を見つけるのが難しい
  - 衝突困難性：h(x)=h(x')となるxとx'の組を見つけるのが難しい

ハッシュ関数が安全でないと？



誕生日のパラドックス

- N人の人が集まっている時、同じ誕生日の人が少なくとも1組ある確率が50%以上になるのはNがいくらのとき？（答え：23人）
- 一般に、ハッシュサイズ（mビット）のハッシュ関数で、h(x)=h(x')となる(x,x')を見つけるのに必要な試行回数は、$2^{m/2}$回
- 従って、$2^{m/2}$回より少ない手間で衝突を見つける方法があれば、そのハッシュ関数は安全でない
- 主なハッシュ関数
  - MD5（ハッシュサイズ128bit）
  - SHA-1(ハッシュサイズ160bit）
  - SHA-2（ハッシュサイズ224,256,384,512bit)

誕生日のパラドックス（導出）

- mビットのハッシュ値がn個あるとき、衝突が起きる確率をPとする。余事象（衝突が一つも起きない）の確率を$Q = (1 - P)$とすると、
- $$Q = \frac{2^m}{2^m} \times \frac{2^m - 1}{2^m} \times \frac{2^m - 2}{2^m} \times \cdots \times \frac{2^m - (n-1)}{2^m}$$
$$= \left(1 - \frac{1}{2^m}\right)\left(1 - \frac{2}{2^m}\right) \cdots \left(1 - \frac{n-1}{2^m}\right)$$
- ここで全ての実数xについて$1 + x \leq e^x$より
$$Q < \prod_{i=1}^{n-1} e^{-\frac{i}{2^m}} = e^{-\frac{1}{2^m}\sum_{i=1}^{n-1} i} = e^{-\frac{1}{2^m} \cdot \frac{n(n-1)}{2}}$$

誕生日のパラドックス（続き）

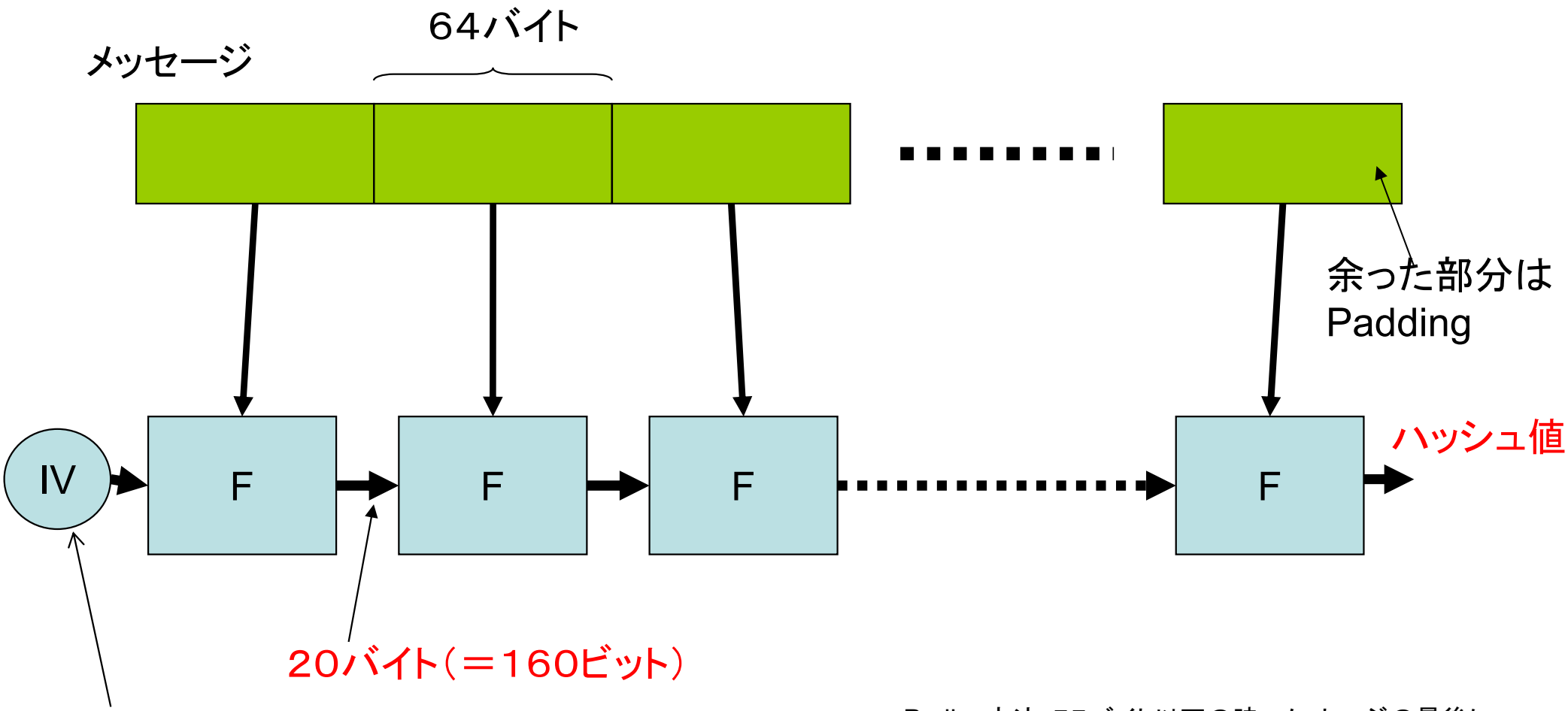
- 従って、$P \geq 1 - e^{-\frac{1}{2^m} \cdot \frac{n(n-1)}{2}}$となる。$P \approx \frac{1}{2}$となる$n$を求めると、$1 - e^{-\frac{1}{2^m} \cdot \frac{n(n-1)}{2}} = \frac{1}{2}$より、
- $n^2 - n - 2 \cdot 2^m \cdot \log_e 2 = 0$を解いて、

$$n = \frac{1}{2}\left(1 + \sqrt{1 + 8 \cdot 2^m \cdot \log_e 2}\right)$$

$$\approx 1.18 \times 2^{\frac{m}{2}}$$

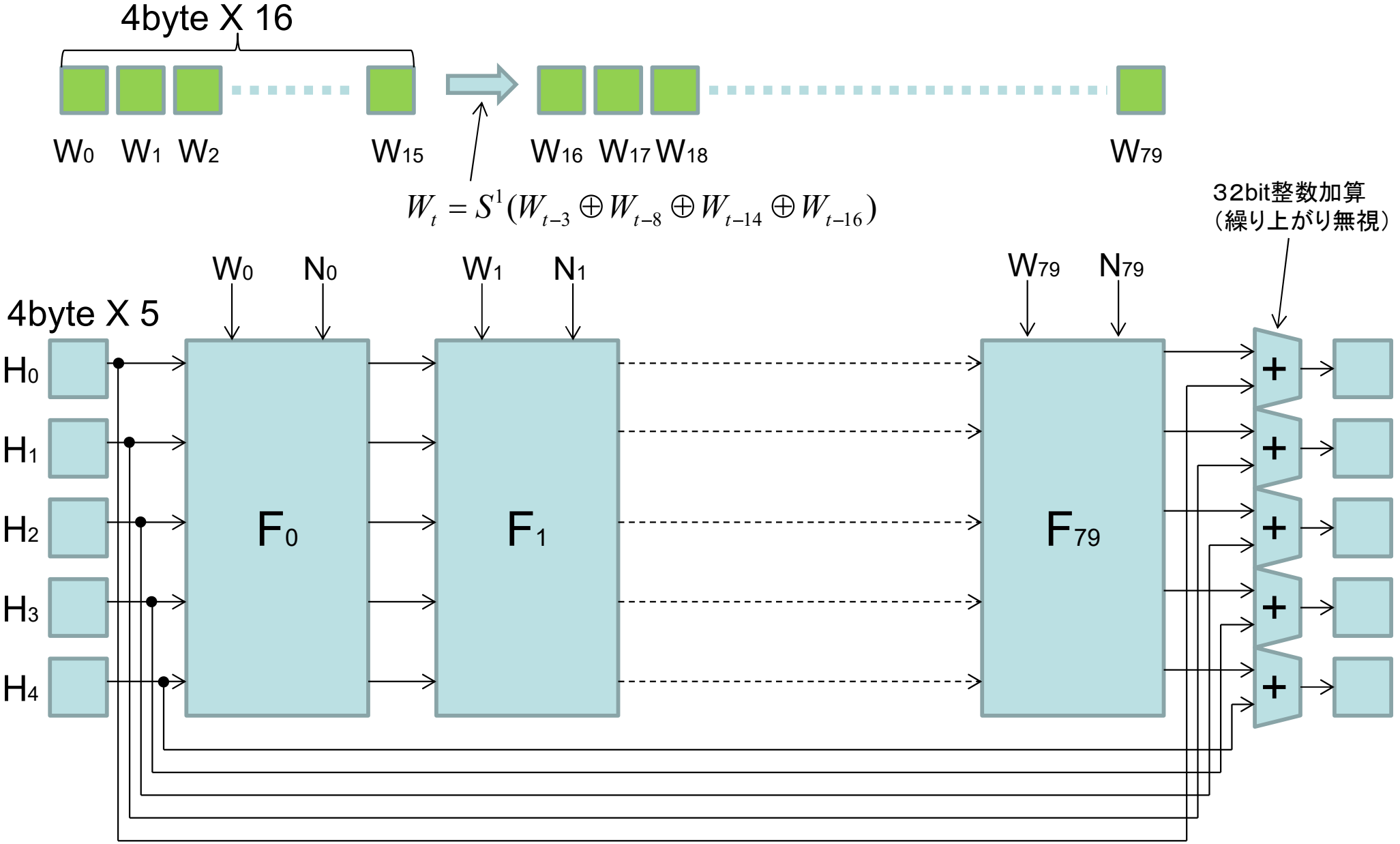
- $\Rightarrow 2^{\frac{m}{2}}$回の試行によりほぼ$\frac{1}{2}$の確率で衝突を見つけることができる。

SHA－１の構造

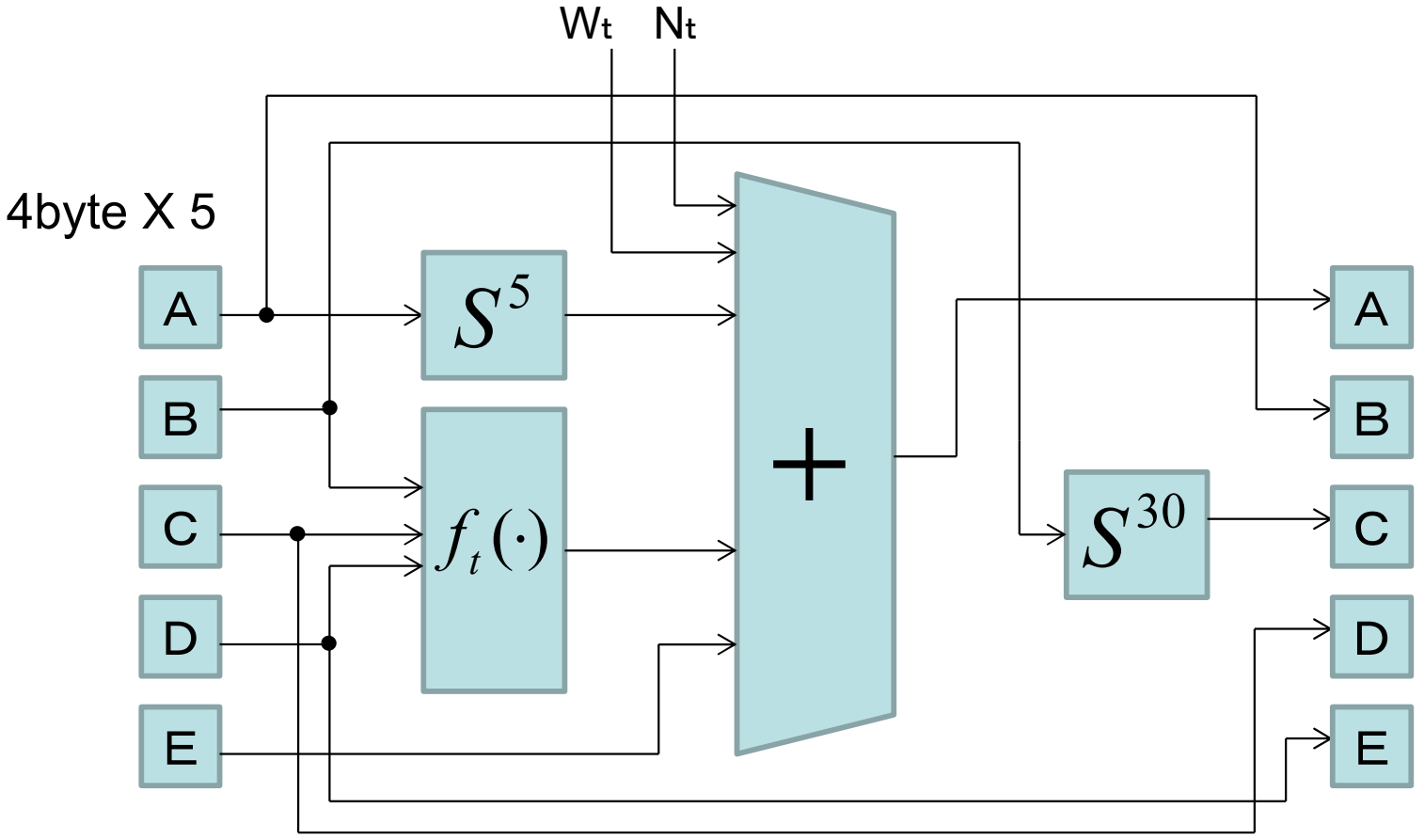


Pading方法：55バイト以下の時、メッセージの最後に80hを付加し残りを0で埋める。ただし最後の8バイトにメッセージの総ビット数を書き込む。
55バイトを超える場合は、メッセージ総ビット数を記録できないので新たなブロックを1つ追加する。

SHA-1の内部構造（1）


4byte X 16
W0 W1 W2 W15 W16 W17 W18 W79
$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$
32bit整数加算
（繰り上がり無視）
4byte X 5
W0 N0 W1 N1 W79 N79
H0 H1 H2 H3 H4
F0 F1 F79
+

SHA－1の内部構造（2）

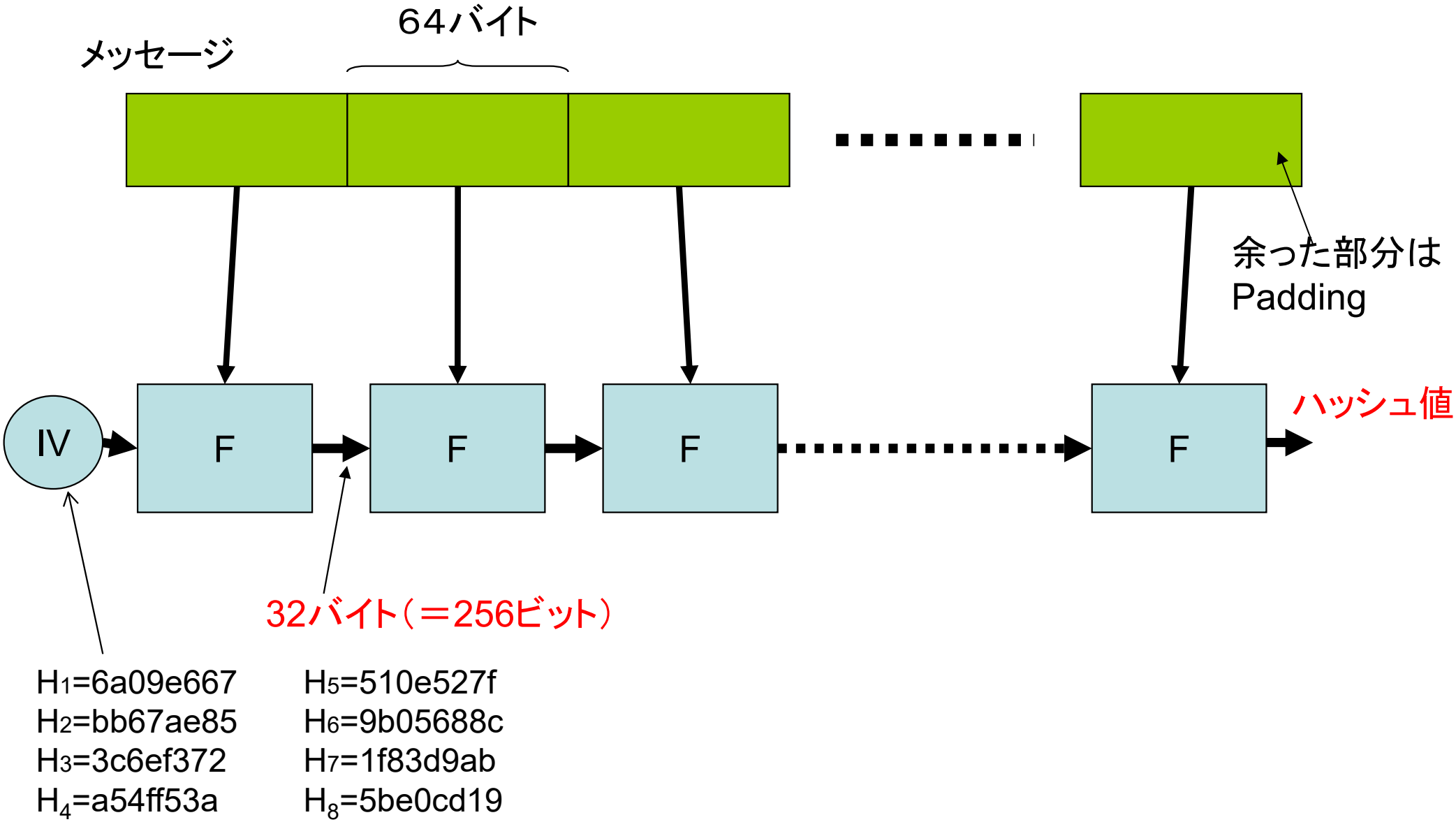


$$f_t(B,C,D) = (B \wedge C) \vee (\overline{B} \wedge D)$$
$$f_t(B,C,D) = B \oplus C \oplus D$$
$$f_t(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$$

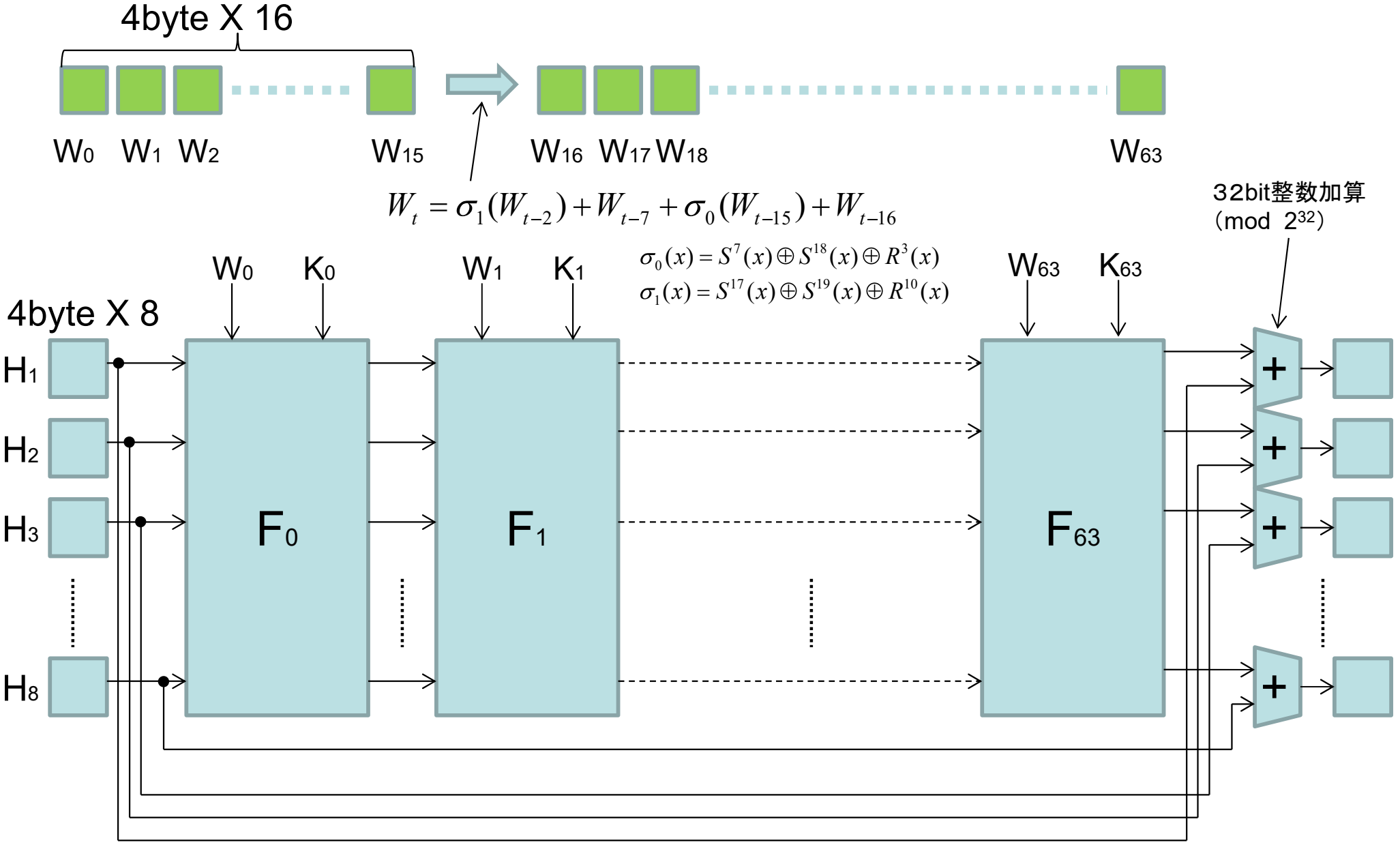
$$0 \le t \le 19$$
$$20 \le t \le 39, 60 \le t \le 79$$
$$40 \le t \le 59$$

$$N_t = \text{5A827999} \quad (0 \le t \le 19)$$
$$N_t = \text{6ED9EBA1} \quad (20 \le t \le 39)$$
$$N_t = \text{8F1BBCDC} \quad (40 \le t \le 59)$$
$$N_t = \text{CA62C1D6} \quad (60 \le t \le 79)$$

SHA－2(SHA-256)の構造

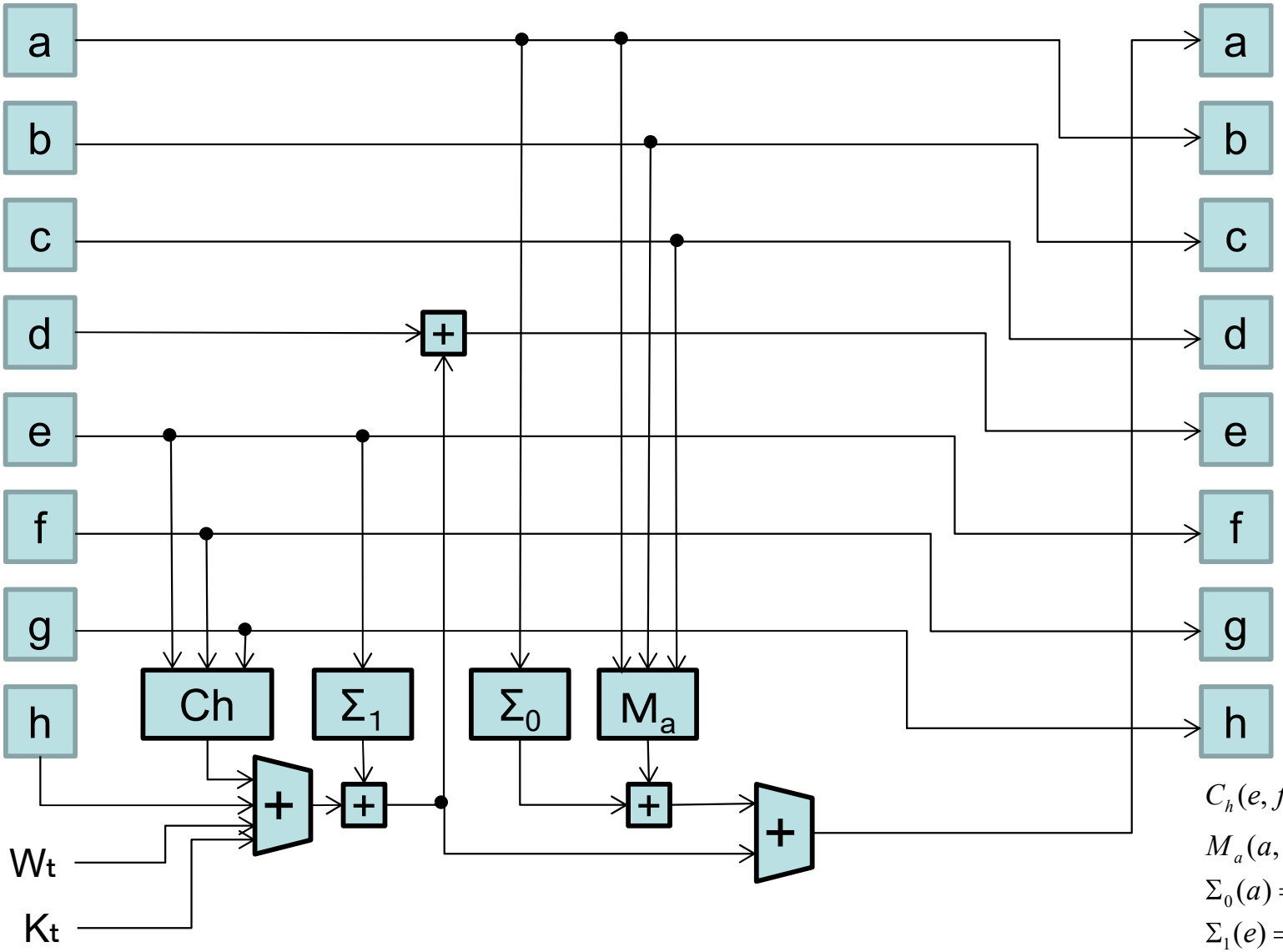


SHA-2の内部構造（1）


4byte X 16
W0 W1 W2 W15 W16 W17 W18 W63
$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$
32bit整数加算
（mod 2^32）
$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x)$
$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)$
4byte X 8
W0 K0 W1 K1 W63 K63
H1 H2 H3 H8
F0 F1 F63
+

SHA－2の内部構造（2）

4byte X 8



$$C_h(e,f,g) = (e \wedge f) \oplus (\bar{e} \wedge g)$$

$$M_a(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$$

$$\Sigma_0(a) = S^2(a) \oplus S^{13}(a) \oplus S^{22}(a)$$

$$\Sigma_1(e) = S^6(e) \oplus S^{11}(e) \oplus S^{25}(e)$$

ディジタル署名の安全性

- ディジタル署名の安全性（偽造のレベル）
  - 一般的偽造不可：署名の偽造ができない文書が存在する
  - 選択的偽造不可：ある決められた文書以外に対しては署名の偽造ができない
  - 存在的偽造不可：どのような文書に対しても署名の偽造ができない
- 攻撃方法の分類
  - 受動攻撃：公開鍵だけを使う攻撃
  - 一般選択文書攻撃：攻撃者が選んだ文書に対する署名をあらかじめ入手できる
  - 適応的選択文書攻撃：攻撃者が選んだ文書に対する署名を入手でき、さらにそれを元に適応的に選んだ文書に対する署名も入手できる
- 「適応的選択文書攻撃に対して存在的偽造不可」が最も厳しい条件。RSA－PSS方式など。

ディジタル署名の偽造の例

- RSA署名
  メッセージm → 署名$\sigma = m^d \pmod n$
- 攻撃者：メッセージmに対する署名を偽造したい
  $m_1 = m \times r$ →署名$\sigma_1 = m_1^{\mathrm{d}} = m^{\mathrm{d}} \times r^{\mathrm{d}} \pmod n$
  $m_2 = r^{-1}$ →署名$\sigma_2 = m_2^{d} = r^{-\mathrm{d}} \pmod n$
  をそれぞれ得る（rは乱数）（選択文書攻撃）

  ↓

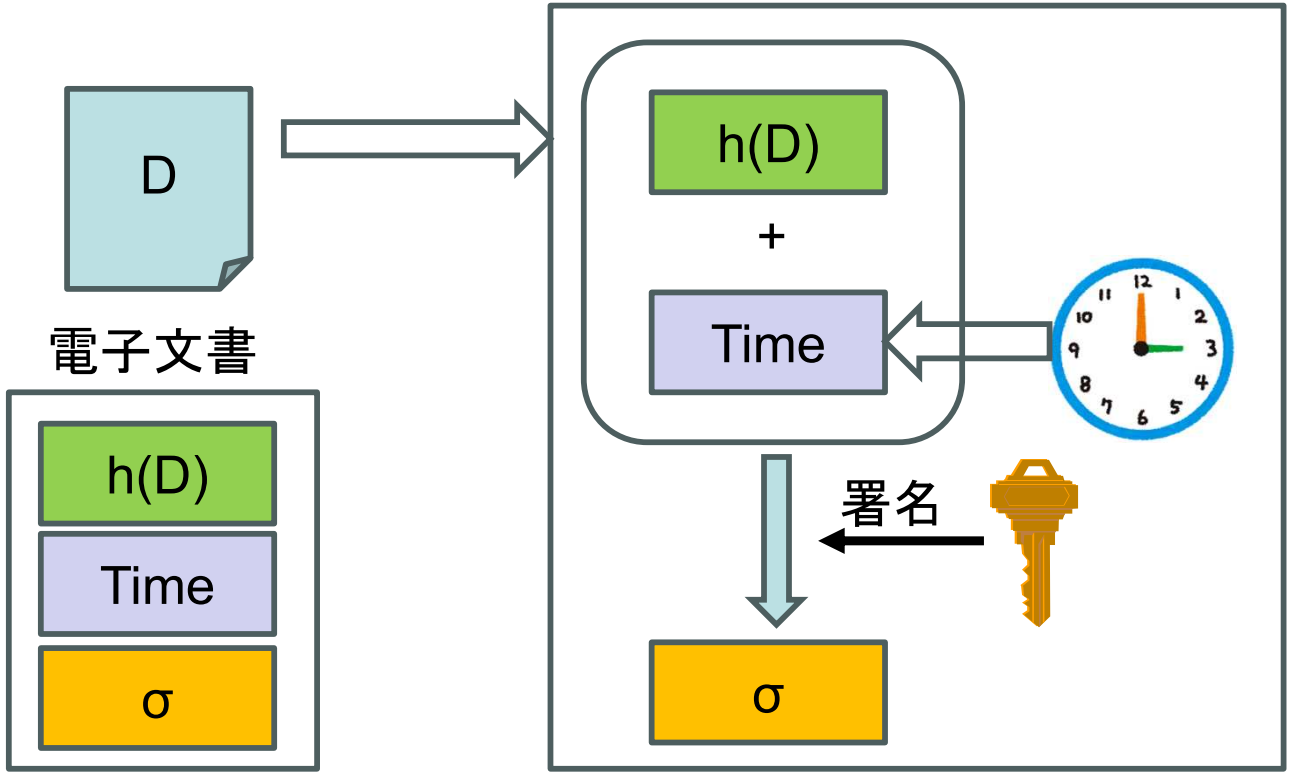
  $\sigma = \sigma_1 \times \sigma_2 = m^d \pmod n$
- 署名$\sigma = \mathrm{h}(m)^d \pmod n$とすれば上の攻撃は不可

電子署名法

- 電子署名及び認証業務に関する法律（2001年4月1日施行）
  - 本人による一定の電子署名がある電子文書は真正に成立したものと推定できる（電子署名を手書き署名や押印と同様に扱う）
  - 認証業務に関する国の認定制度が導入された

タイムスタンプ

- 電子署名によって、その文書が誰によって作成されたのか（その後、改ざんされていないか）は証明できるが、いつ作成されたのかは証明できない→電子文書が成立した日時を特定する必要性



タイムスタンプ局
(Time Stamping Authority: TSA)

e-文書法に対応するために用いるタイムスタンプは、（財）日本データ通信協会タイムビジネス認定センターに認定されたTSAにより発行されたものであることが求められる