Hash Function

$H : M \to T$

$H \to$ random oracle

- Coll in $H \Rightarrow \sqrt{T}$ queries
- PI in H, 2PI in H $\Rightarrow T$ queries

1. Hash function from compression function
2. Hash function from public permutation.
3. Applications.

Hash Function

$H : M \to T$

$H \to$ random oracle

- Coll in $H \Rightarrow \sqrt{T}$ queries
- PI in H, 2PI in H $\Rightarrow T$ queries

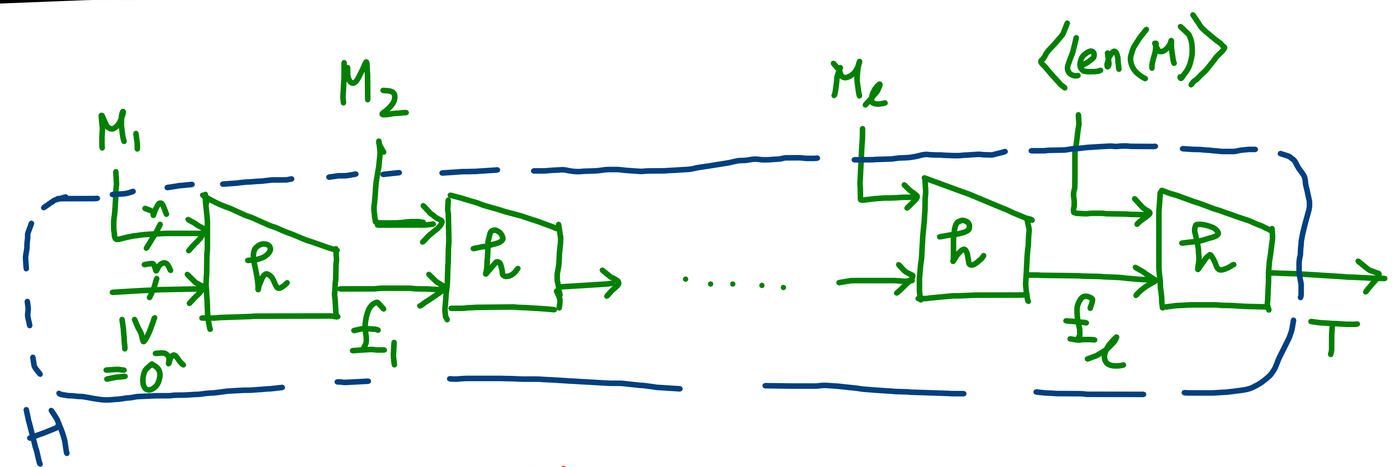
1. Hash function from compression function
2. Hash function from public permutation.
3. Applications.

Hash function from compression functions

Merkle - Damgard

$h: \{0,1\}^{2n} \to \{0,1\}^n$

$H: \{0,1\}^* \to \{0,1\}^n$



M

$M_1 \| M_2 \| \cdots \| M_\ell \leftarrow Pad(M)$, $|M_i| = n$

$\{0,1\}^{200} \Rightarrow \ell = 2$

$n = 128$ $\quad len(M) = 200$

$\Downarrow$

$Pad(M) = M \| 0^{56}$

Claim: If h is collision-resistant, then H is collision-resistant.

Assume coll in H $\quad$ so is h.

↳ $M, M^* \Rightarrow H(M) = H(M^*)$

Case 1: $\langle len(M) \rangle \neq \langle len(M^*) \rangle$

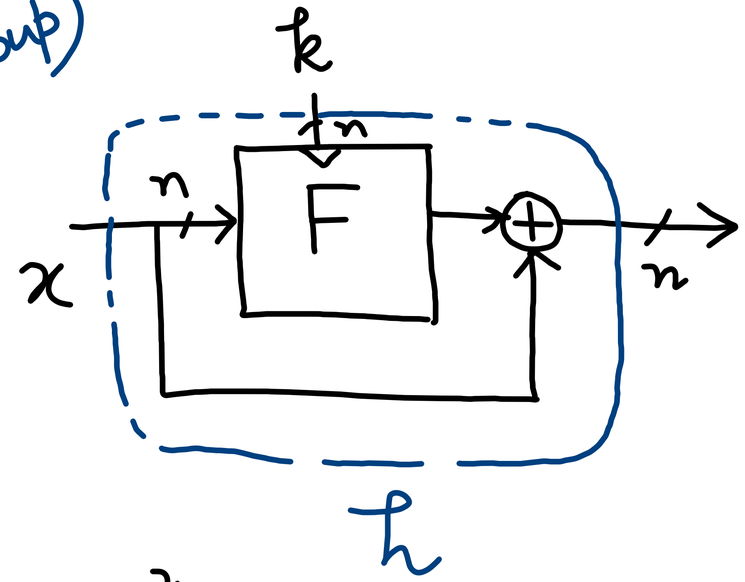
$h(\langle len(M) \rangle, f_\ell) = h(\langle len(M^*) \rangle, f_\ell^*) \Rightarrow$ Collision in h

Case 2: $\langle len(M) \rangle = \langle len(M^*) \rangle$

$\exists i,\ h(M_i, f_{i-1}) = h(M_i^*, f_{i-1}^*)$ s.t. $(M_i, f_{i-1}) \neq (M_i^*, f_{i-1}^*)$

Davies-Meyer Compression Function

Ref (Boneh-Shoup)



If F is modeled as ideal cipher, DM construction is collision resistant up to $2^{n/2}$ queries.

$h : \{0,1\}^{2n} \to \{0,1\}^n$

Coll $\to (x_1, k_1)\ (x_2, k_2) \Rightarrow \frac{\binom{q}{2}}{2^n}$

$F_{k_1}(x_1) \oplus F_{k_2}(x_2) = x_1 \oplus x_2$

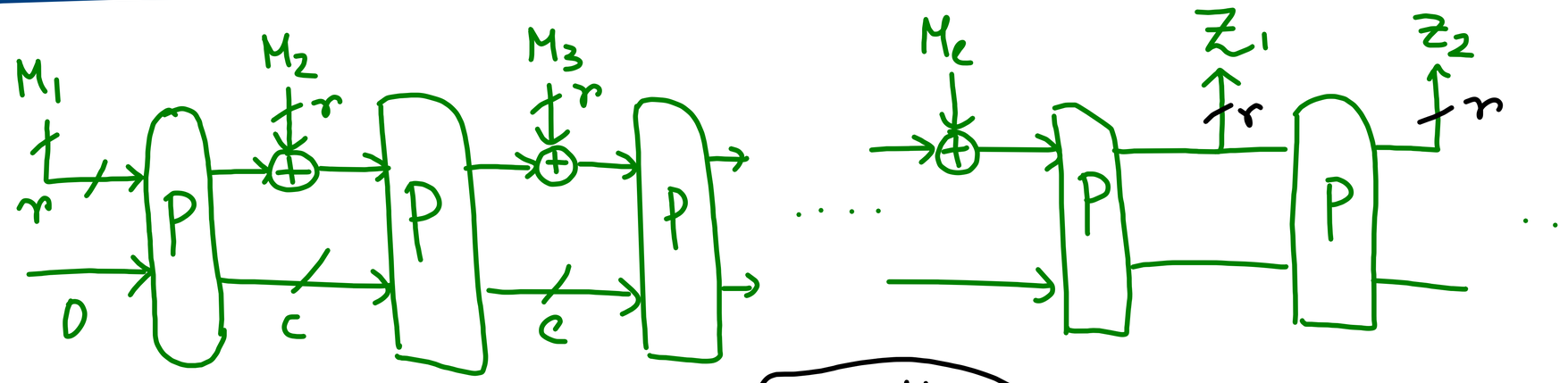
$$\left.\begin{aligned} F(k_1, m_1) &= y_1 \\ F(k_1, m_2) &= y_2 \\ F(k_2, m_1) &= y_3 \end{aligned}\right\} \textcircled{3}$$

$$Pr\left[F(k_2, m_2) = y_4 \mid \textcircled{3}\right] = \frac{1}{2^n}$$

Hash Function from Public Permutation

$$P: \{0,1\}^{r+c} \rightarrow \{0,1\}^{r+c}$$

Sponge Mode



$M_1 \| M_2 \| \cdots \| M_l = Pad(M)$

$|M_i| = r$

↳ 10^* padding

If full block is not padded

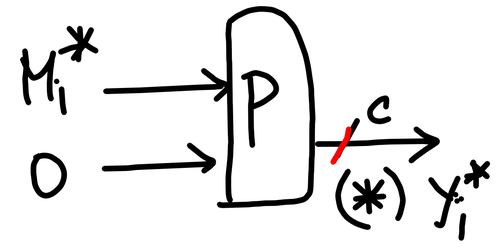
$H(0^{127}) = H(0^{127}1)$.

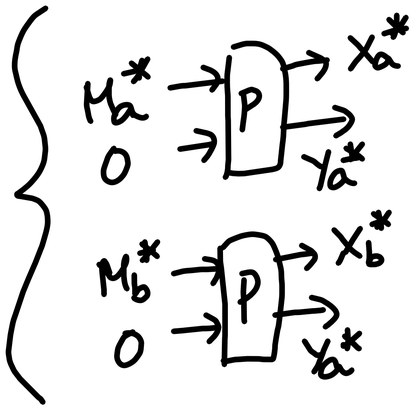
Hash size = $2r$

$\left\lceil \frac{n}{r} \right\rceil$

P → random permutation

Collision resistant up to $2^{c/2}$ queries





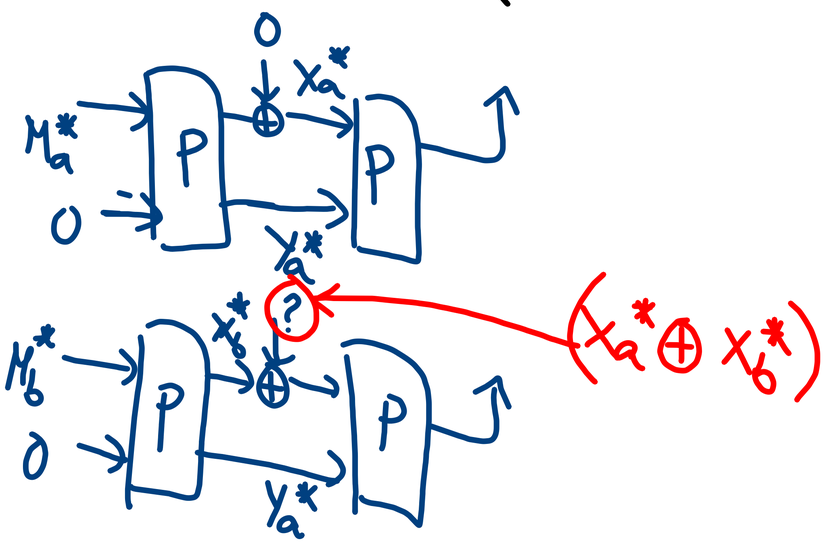
$$Y_a^* = Y_b^*$$

$$M_1^* \to y_1^*$$
$$M_2^* \to y_2^*$$
$$\vdots$$
$$M_q^* \to y_q^*$$

$$Pr[\exists\, i, j,\ y_i^* = y_j^*] \simeq \frac{\binom{q}{2}}{2^c}$$

(Collision in the capacity part $\Rightarrow$ Collision in the hash)

$$H(M_a^* \| 0) = H(M_b^* \|$$



Applications

$X_1 \rightarrow$ Stored in cloud

How to check integrity

$H(x_1) \rightarrow$ Store locally

- Store $h(x_1)$

Client — Server

x_1 →

← request for file

← x_1'

If $h(x_1) = h(x_1')$
File integrity

$x_1 \rightarrow$
⋮
$x_{2^{30}} \rightarrow$

$x' \rightarrow \boxed{h_1, \ldots, h_{2^{30}}}$

$h(x') \rightarrow$ Store

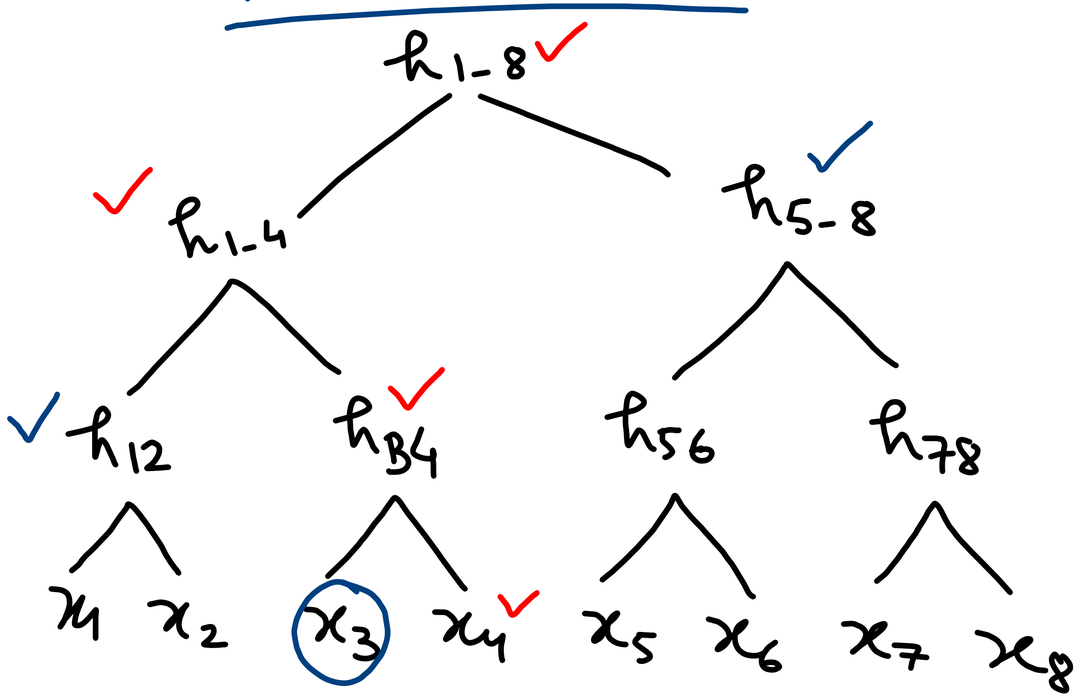
x_i →

← $x_1, \ldots, x_{2^{30}}$

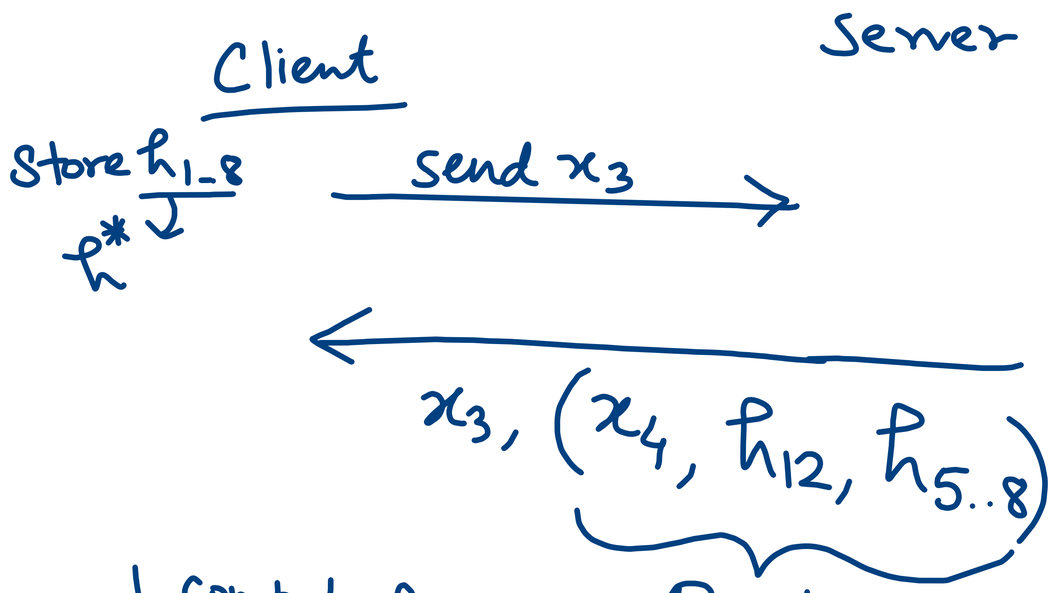
⇓

Then only we can check.

Merkle tree

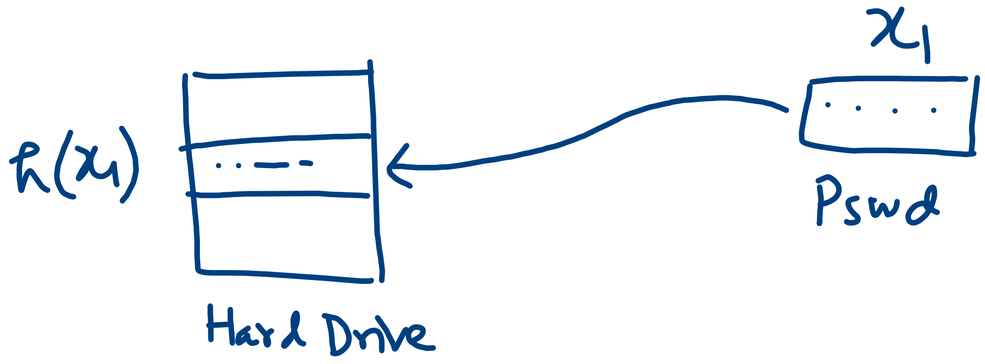


$n = 2^k \Rightarrow (k-1)$ many hash values + 1 additional file.

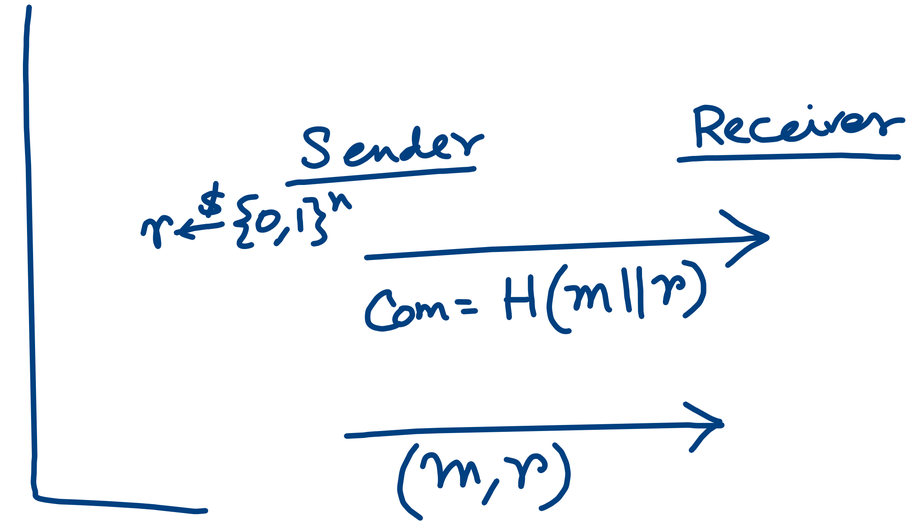


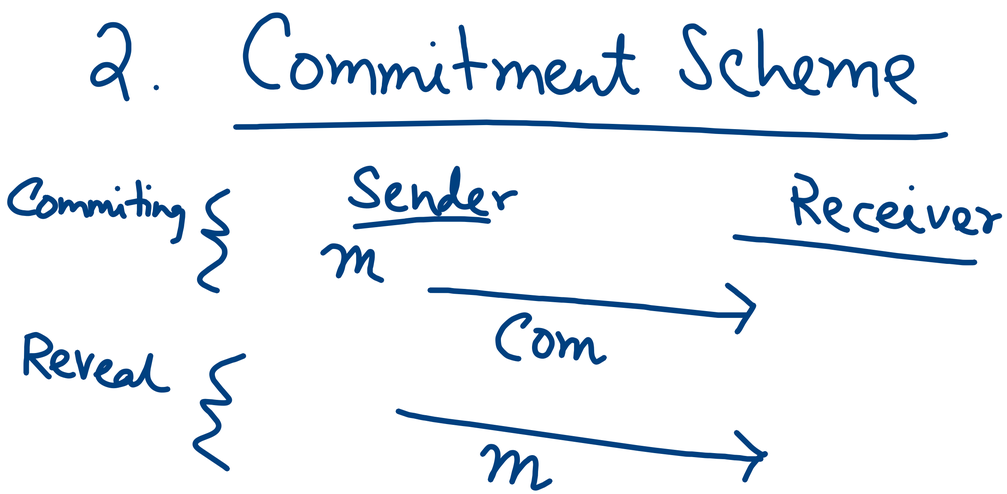
1. compute h_{34}, $h_{1..4}$, h_{1-8}
2. Compare h_{1-8} with h^*

1. Password Hashing



2. Commitment Scheme





1. Hiding Property
   - ↳ Com should never reveal m.
2. Binding
   - ↳ Com should ensure that the underlying message was only M.

0^* padding

10^* (optional)

↳ Full block → no padding

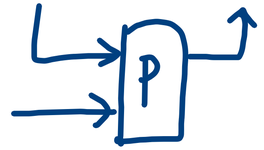
↳ Partial block → 10^*

$M_1 = 0^{126}$

$M_2 = 0^{127}$ ⇒

$H(M_1) = H(M_2)$

0^{128}



⇐

$\forall M_1, M_2$ **Prop 1**

$M_1 \neq M_2$

$Pad(M_1) \neq Pad(M_2)$

$M_1 = 0^{127}$

$M_2 = 0^{127}1$ $\quad Pad(M_1) = Pad(M_2)$

Always 10^* padding

⇒ Prop 1