Cryptographic Hash Functions
Problem Set 2

Summer 2019

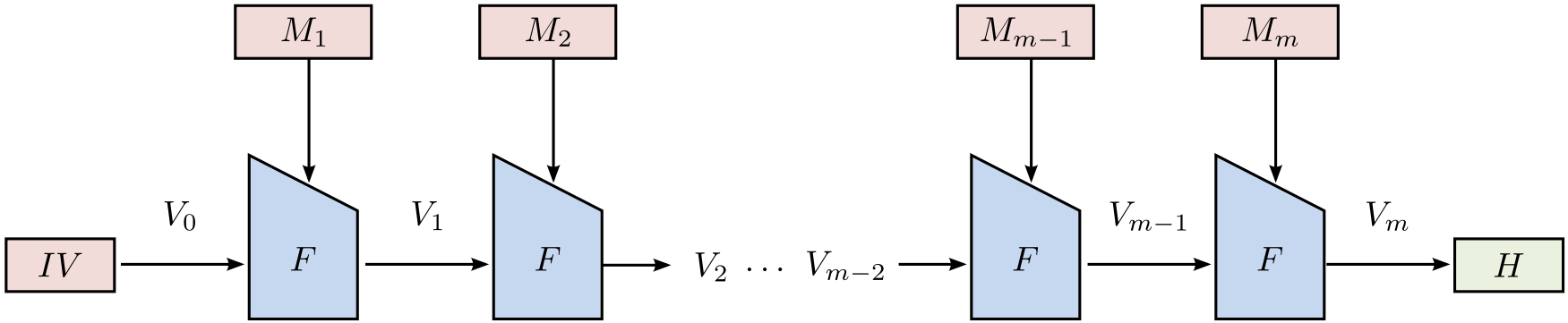
Chair of Media Security Prof. Stefan Lucks
Jannis Bossert, Nathalie Dittrich, Eik List <firstname>.<lastname>(at)uni-weimar.de

Bauhaus-Universität Weimar

May 2, 2019

Recap: The Merkle-Damgård Paradigm

- Given: Compression function $F : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^\ell$
- Goal: build a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$
  - Append message length and pad M: $M \leftarrow M \,||\, |M| \,||\, 10^*$
  - Split $M_1, \ldots, M_m \xleftarrow{n} M$ into n-bit blocks
  - Iteratively compute $V_i \leftarrow F(V_{i-1}, M_i)$ for $i \in 1, \ldots, m$
  - $V_0 := IV$ is a public constant IV
  - $V_m := H(M)$ is the hash



Task 1: Proof of Storage

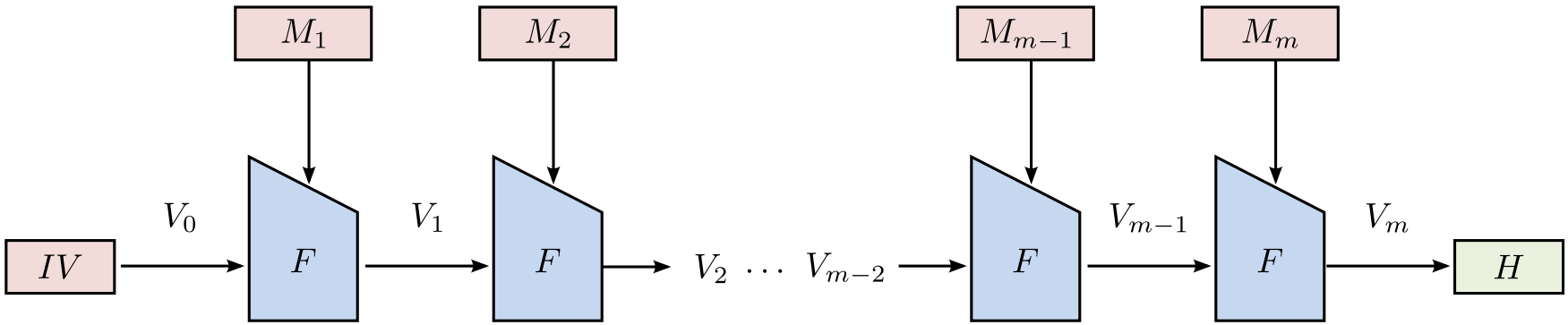
- Game between Challenger Alice and Adversary Eve
- Eve has limited resources bounded by $O(n^c)$ for some constant c
- Alice samples a challenge $M \leftarrow\!\!\!\!\!\!\leftarrow \mathcal{M}$, with $\mathcal{M} \subseteq \{0,1\}^n$ and submits it to Eve
- Alice and Eve agree on a verification algorithm $F : \mathcal{M} \times \mathcal{N} \rightarrow \mathcal{V}$
- Eve wins if she can efficiently compute the output of F while storing only $O(n^c)$ of M

Task 1: Proof of Storage

1. Alice sends a random n-bit challenge N to Eve. Eve responds with $Y = H(M \,||\, N)$.
2. Alice sends a random n-bit challenge N to Eve. Eve responds with $Y = H(N \,||\, M)$.

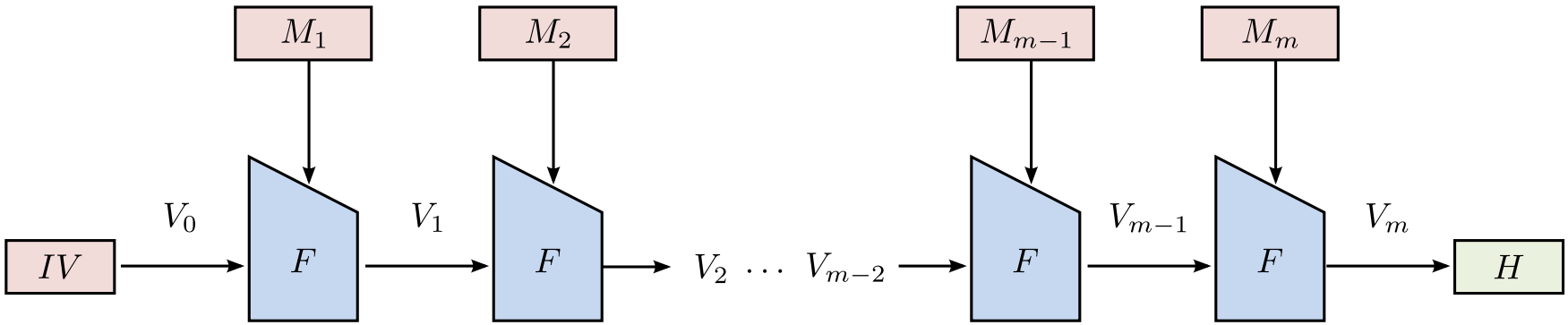
Does Alice really have to store M?

Task 2: Proving Merkle-Damgård



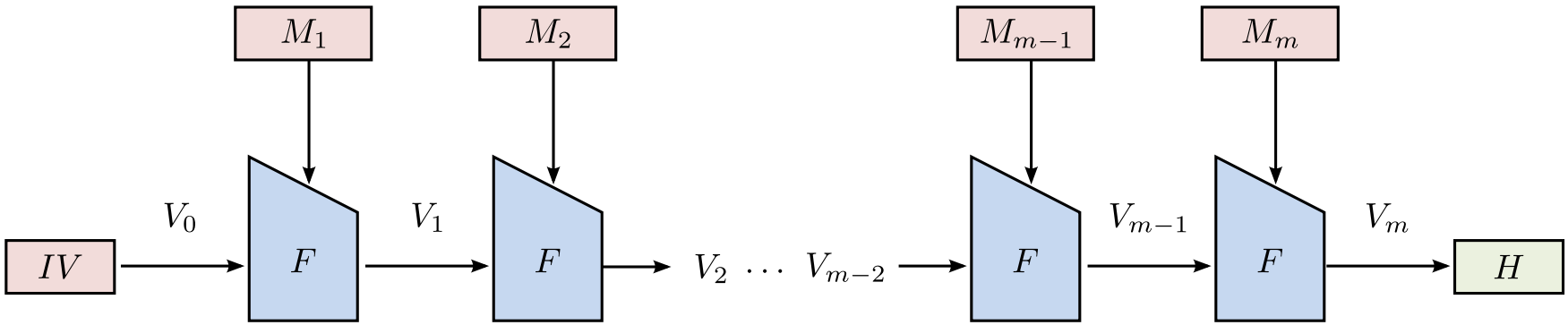
- **Goal:** Show that if F is collision-resistant $\implies$ H is collision-resistant

Task 2: Proving Merkle-Damgård



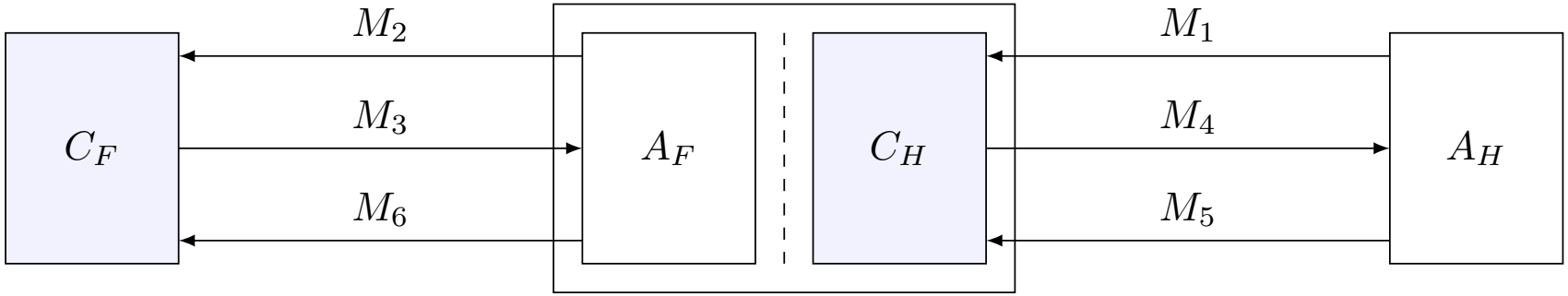
- **Goal:** Show that if F is collision-resistant $\implies$ H is collision-resistant
- **Proof by contradiction:** If there exists an efficient collision-finding adversary $\mathcal{A}_H$ on H $\implies$ we can use it to construct an efficient collision-finding adversary $\mathcal{A}_F$ on F

Task 2: Proving Merkle-Damgård



- **Goal:** Show that if F is collision-resistant $\implies$ H is collision-resistant
- **Proof by contradiction:** If there exists an efficient collision-finding adversary $\mathcal{A}_H$ on H $\implies$ we can use it to construct an efficient collision-finding adversary $\mathcal{A}_F$ on F
- Assume two messages M, M'

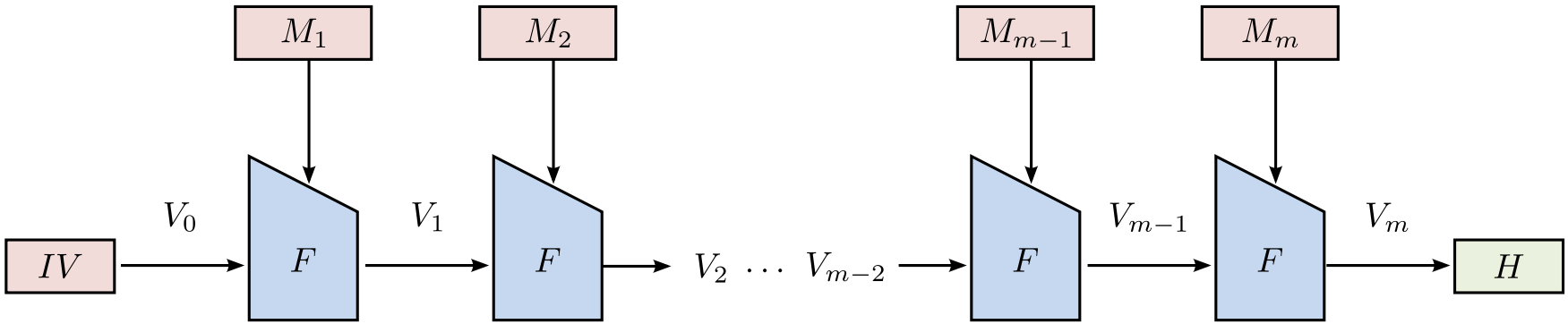
Task 2: Proving Merkle-Damgård



- $\mathcal{A}_F$ interacts with compression-function challenger $\mathcal{C}_F$
- $\mathcal{A}_F$ can ask (V^i_{j-1}, M^i_j) and obtains $V^i_j \leftarrow F(V^i_{j-1}, M^i_j)$
- End: $\mathcal{A}_F$ has to provide distinct (V_{c-1}, M_c) and (V'_{c-1}, M'_c) and wins if $F(V_{c-1}, M_c) = F(V'_{c-1}, M'_c)$
- $\mathcal{A}_F$ simulates $\mathcal{A}_H$ by acting as a hash-function challenger $\mathcal{C}_H$
- $\mathcal{A}_H$ can ask M^i and obtains $V^i_0 = H(M^i)$
- $\mathcal{A}_F$ uses its $\mathcal{C}_F$ oracle
- End: $\mathcal{A}_H$ has to provide distinct (M, M') and wins if $H(M) = H(M')$
- To show: $\mathcal{A}_F$ extracts collision (V_{c-1}, M_c) from M and (V'_{c-1}, M'_c) from M' and passes it to $\mathcal{C}_F$

Task 2: Proving Merkle-Damgård

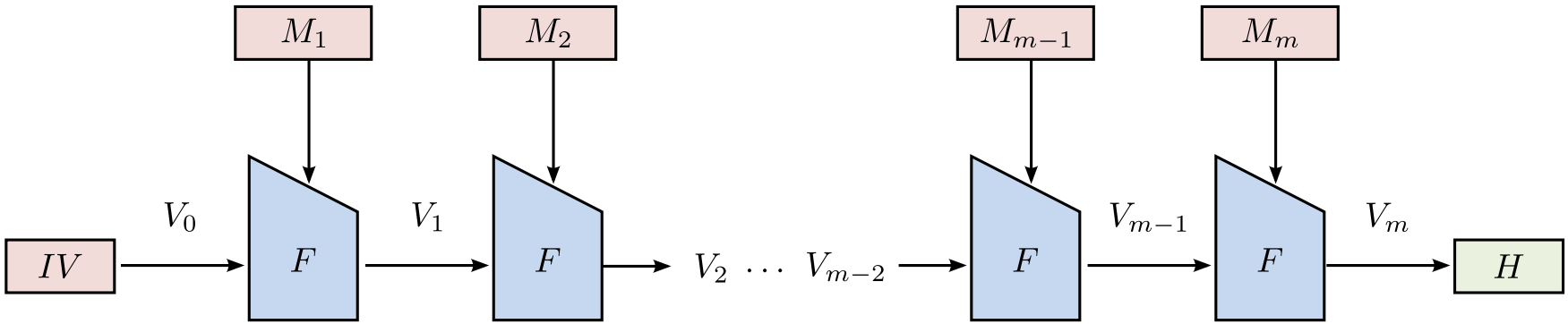
Cases



- Assume $M = (M_1, \ldots, M_m) \neq M' = (M'_1, \ldots, M'_{m'})$ is valid collision for H found by $\mathcal{A}_H$
- WLOG.: $m \geq m'$
- There must exist some block $0 \leq c \leq m'$ s.t. $V_c = V'_c$.
- We consider three mutually exclusive possible cases
- **Case 1: $\mathbf{m > m'}$.**

Task 2: Proving Merkle-Damgård

Cases

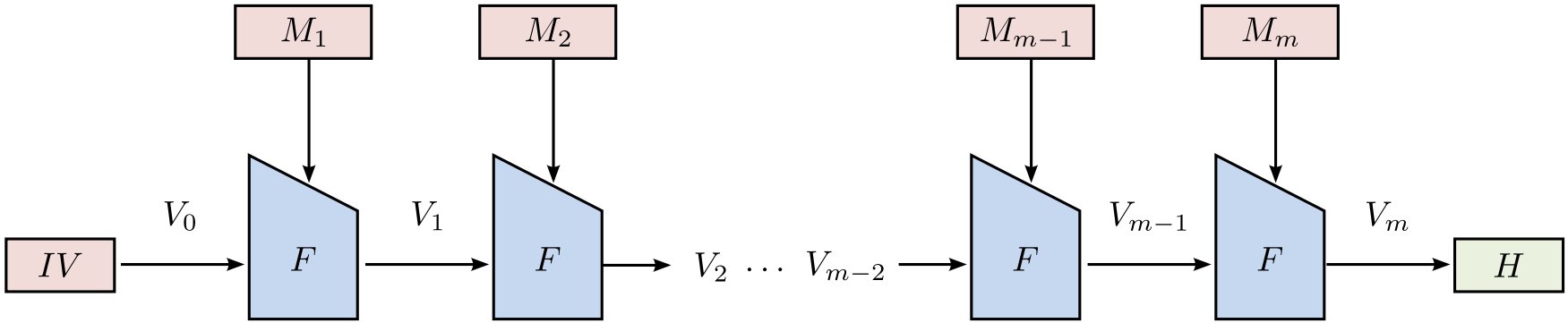


- Assume $M = (M_1, \ldots, M_m) \neq M' = (M'_1, \ldots, M'_{m'})$ is valid collision for H found by $\mathcal{A}_H$
- WLOG.: $m \geq m'$
- There must exist some block $0 \leq c \leq m'$ s.t. $V_c = V'_c$.
- We consider three mutually exclusive possible cases
- **Case 1: $\mathbf{m > m'}$.** The length encoding ensures $M_m \neq M'_{m'}$. $\mathcal{A}_H$ must find a collision for $F(V_m, M_m) = F(V'_m, M'_{m'})$:

$$\mathbf{Adv}_H^{\textsc{Coll}}(\mathcal{A}_H) \leq \mathbf{Adv}_F^{\textsc{Coll}}(\mathcal{A}_F).$$

Task 2: Proving Merkle-Damgård

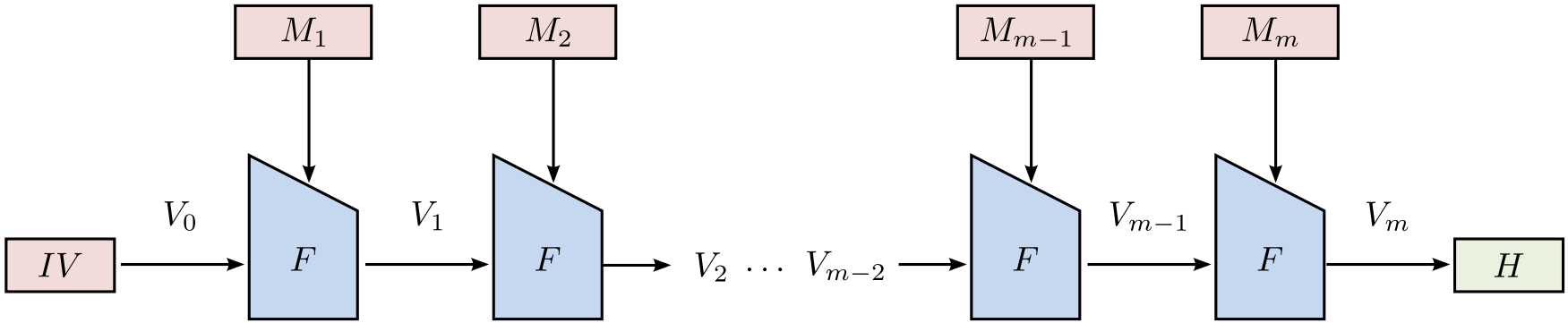
Cases



- **Case 2: $\mathbf{m = m' \wedge c > 1}$.**

Task 2: Proving Merkle-Damgård

Cases



- **Case 2: $\mathbf{m = m' \wedge c > 1}$.** $\mathcal{A}_H$ must have found a collision for the compression function

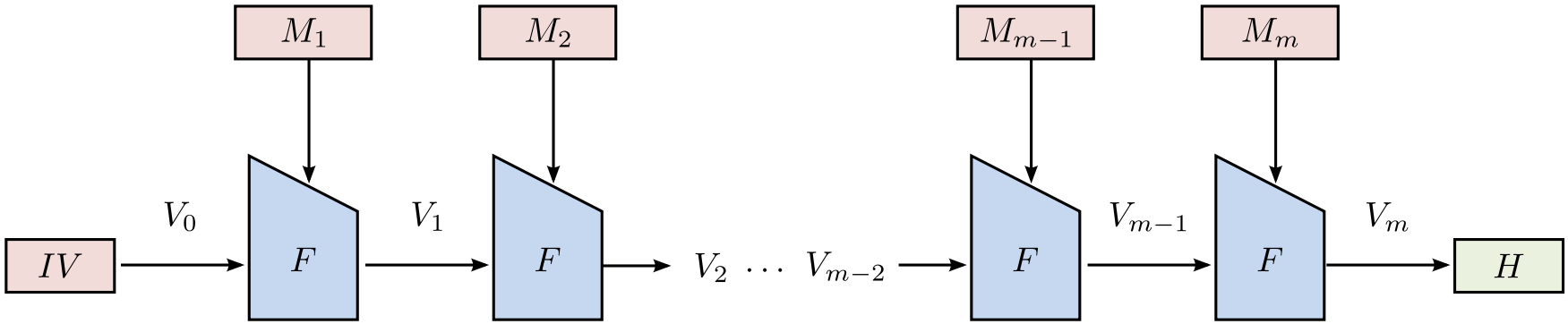
$$F(V_{c-1}, M_c) = F(V'_{c-1}, M'_c) = V_c.$$

Advantage is

$$\mathbf{Adv}_H^{\text{Coll}}(\mathcal{A}_H) \leq \mathbf{Adv}_F^{\text{Coll}}(\mathcal{A}_F).$$

Task 2: Proving Merkle-Damgård

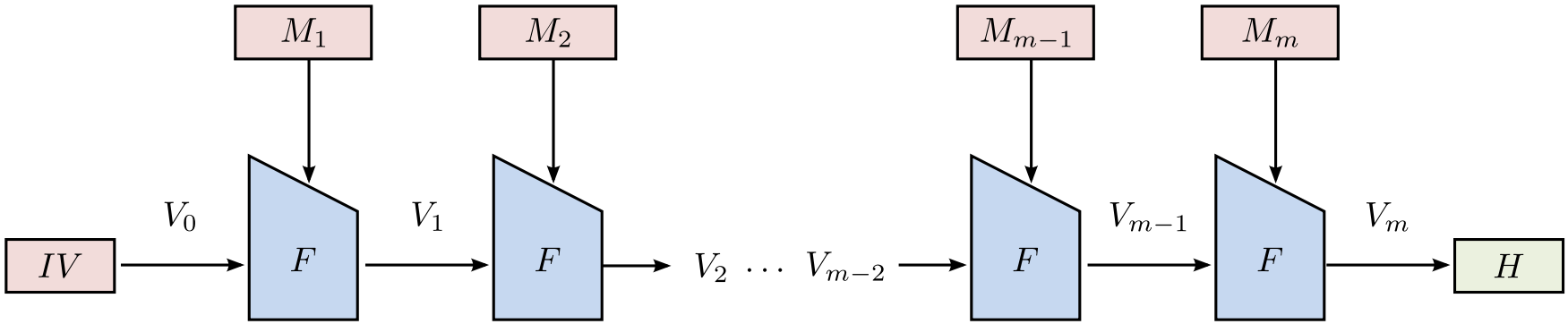
Cases



- **Case 3: $\mathbf{m = m' \wedge c = 1}$.**

Task 2: Proving Merkle-Damgård

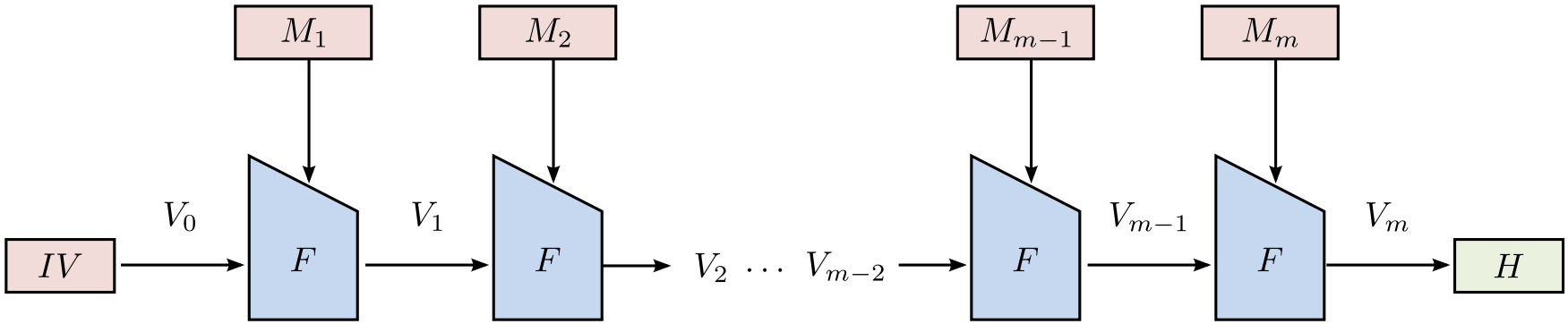
Cases



- **Case 3: $\mathbf{m = m' \wedge c = 1}$.** Represents a collision $V_1 = V_1' = IV$, which means that the IVs collide, which is trivial.

Task 2: Proving Merkle-Damgård

Cases

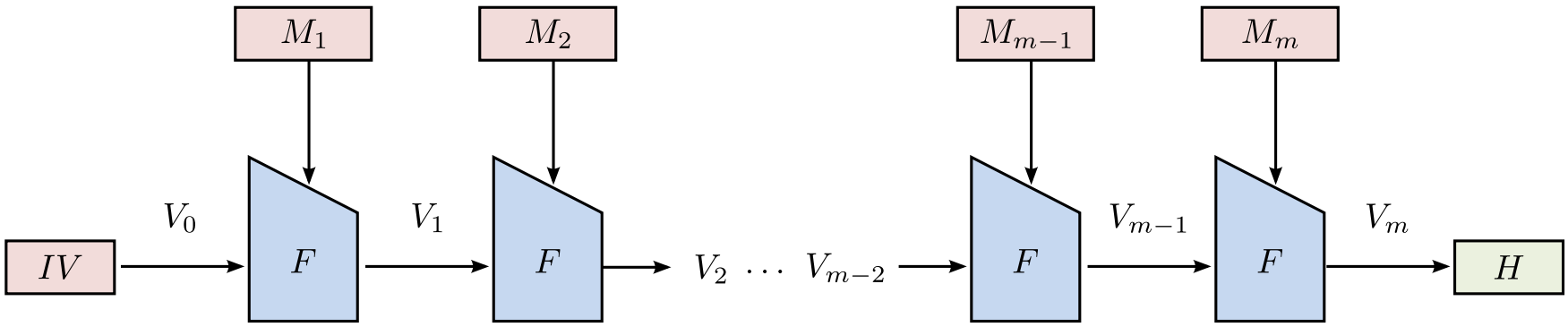


- **Case 3: $\mathbf{m = m' \wedge c = 1}$.** Represents a collision $V_1 = V_1' = IV$, which means that the IVs collide, which is trivial.

Since the cases cover all possibilities and are mutually exclusive:

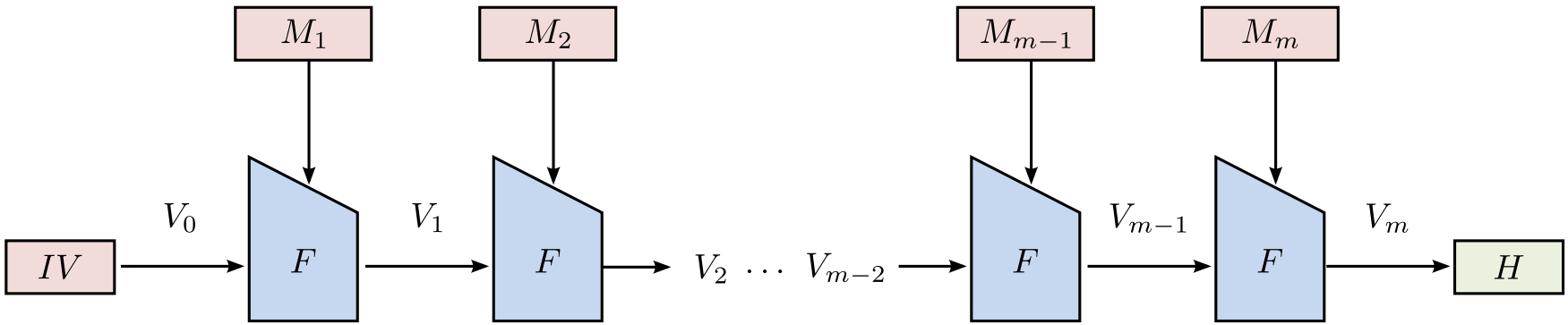
$$\mathbf{Adv}_H^{\mathcal{A}_H}(q_H, \sigma_H) \leq \mathbf{Adv}_F^{\mathcal{A}_F}(q_H + \sigma_H).$$

Task 3: Joux' Multi-Collisions



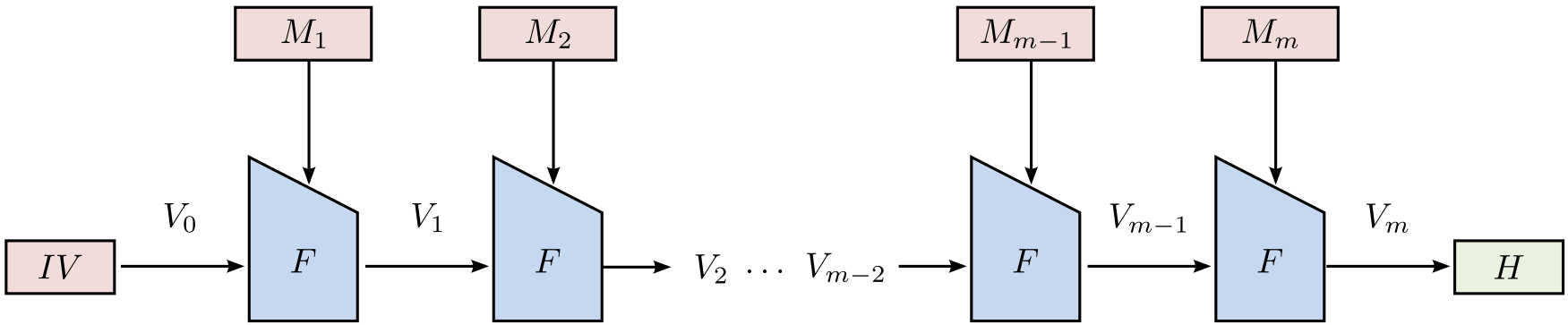
- For $c = 1 \ldots 2$:

Task 3: Joux' Multi-Collisions



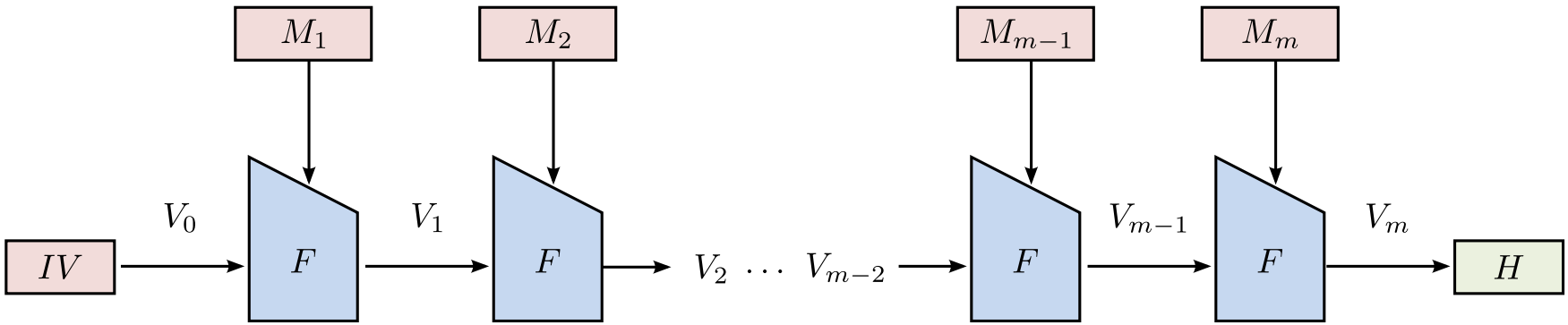
- For $c = 1 \ldots 2$:
  - Query M_1^i until $M_1 = M_1'$ is found ($2^{n/2}$ queries)

Task 3: Joux' Multi-Collisions



- For $c = 1 \ldots 2$:
  - Query M_1^i until $M_1 = M_1'$ is found ($2^{n/2}$ queries)
  - Query M_2^i until $M_2 = M_2'$ is found ($2^{n/2}$ queries)

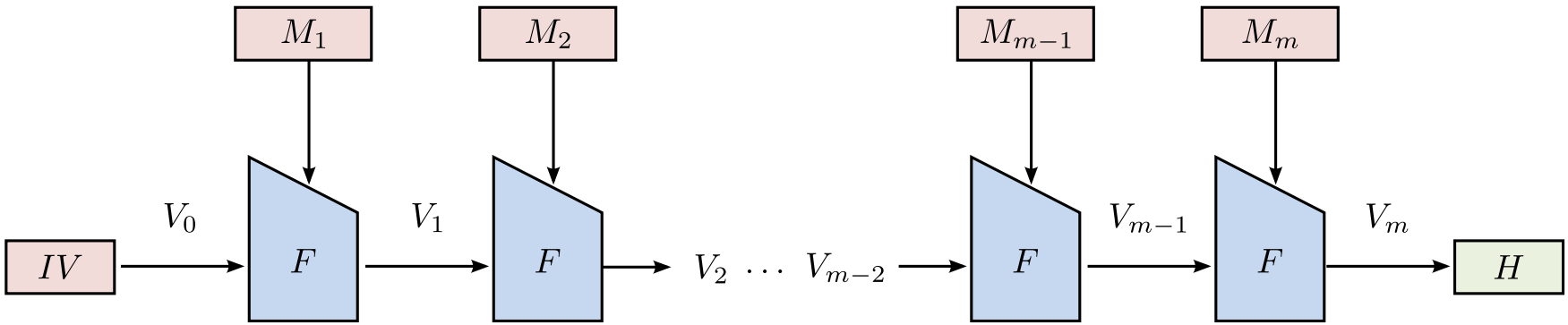
Task 3: Joux' Multi-Collisions



- For $c = 1 \ldots 2$:
  - Query M_1^i until $M_1 = M_1'$ is found ($2^{n/2}$ queries)
  - Query M_2^i until $M_2 = M_2'$ is found ($2^{n/2}$ queries)
- We have a 4-collision:

$$H(M_1, M_2) = H(M_1', M_2) = H(M_1, M_2') = H(M_1', M_2')$$

Task 3: Joux' Multi-Collisions



- For $c = 1 \ldots 2$:
  - Query M_1^i until $M_1 = M_1'$ is found ($2^{n/2}$ queries)
  - Query M_2^i until $M_2 = M_2'$ is found ($2^{n/2}$ queries)
- We have a 4-collision:

$$H(M_1, M_2) = H(M_1', M_2) = H(M_1, M_2') = H(M_1', M_2')$$

- For general $c \geq 1$: $\mathcal{O}(c \cdot 2^{n/2})$ effort for a 2^c-collision

Task 4: Birthday Paradox

Throwing q balls randomly and independently into n bins. What is the probability that two balls fall into the same bin:

$$\begin{aligned}
\Pr[\textsc{Coll}] &= 1 - \frac{n!}{(n-q)! \cdot n^q} \\
&\approx 1 - \left(\frac{n}{n-q}\right)^{n-q+0.5} \cdot e^{-q} \\
\Pr[\textsc{Coll}] &= 1 - \prod_{i=1}^{q-1} \left(1 - \frac{i}{n}\right) \\
&\approx \prod_{i=1}^{q} e^{-\frac{i}{n}} \approx e^{-\sum_{i=1}^{q-1} \frac{i}{n}} \approx e^{-q^2/(2n)}
\end{aligned}$$

Remember: Taylor series approximation: $e^x \approx 1 + x$, for $x \ll 1$

Task 4: Birthday Paradox

Let $H : \{0,1\}^{256} \to \{0,1\}^{256}$ be a hash function with the following property:

$$w(x) = w(H(x)), \qquad \text{for all } x \in \{0,1\}^{256}, \tag{1}$$

where $w(x) := \sum_{i=0}^{n-1} x_i$ is the hamming weight.

Task 4: Birthday Paradox

For all $i \leq n$, H defines a set of $n+1$ functions

$$H := \{H_i : \mathcal{D}_i \to \mathcal{R}_i\}_{0 \leq i \leq n},$$

with pair-wise disjoint domains $\mathcal{D}_i := \{x \in \{0,1\}^{256} | \, w(x) = i\}$ and ranges $\mathcal{R}_i := \{x \in \{0,1\}^{256} | \, w(x) = i\}$.
Number of elements in classes:

$$|\mathcal{D}_i| = |\mathcal{R}_i| = \binom{256}{i}$$

- $\binom{256}{0} = \binom{256}{256} = 1$ element for $i \in \{0, 256\}$
- $\binom{256}{1} = \binom{256}{255} = 256$ elements for $i \in \{1, 255\}$
- $\binom{256}{2} = \binom{256}{254} = 256 \cdot 255/2$ elements for $i \in \{2, 254\}$
- ...

Assumption: Each H_i maps elements $x \in \mathcal{D}_i$ indendently and uniformly at random to $y \to \mathcal{R}_i$

Task 4: Birthday Paradox

a) Find a collision on H and the complexities for a success probability of 0.5 and 0.99, respectively.

Task 4: Birthday Paradox

a) Find a collision on H and the complexities for a success probability of 0.5 and 0.99, respectively.

- No collisions for $w(x) = 0$ or $w(x) = n$
- Next highest: $w = 1$ or $w = 255$ (2^{-8} for a collision)

#Queries	$\Pr[Coll]$	$\Pr[Coll]$ (Taylor)
16	0.3802923025	0.3934693402
17	0.4190240335	0.4313289462
18	0.4576044688	0.4689040089
19	0.4957416546	0.5059300264
20	0.5331670787	0.5421666382
21	0.5696384006	0.5773995567
...	...	...
47	0.9889400452	0.9866261386
48	0.9909705837	0.9888910034
49	0.9926635993	0.9908082885

Task 4: Birthday Paradox

b) Find a preimage $x \in \{0,1\}^{256}$ to a hash value $y \in \{0,1\}^{256}$, chosen by the adversary, and provide the lowest possible complexity.

Task 4: Birthday Paradox

b) Find a preimage $x \in \{0,1\}^{256}$ to a hash value $y \in \{0,1\}^{256}$, chosen by the adversary, and provide the lowest possible complexity.

- Choose $y \in \{0^{256}, 1^{256}\}$

Task 4: Birthday Paradox

b) Find a preimage $x \in \{0,1\}^{256}$ to a hash value $y \in \{0,1\}^{256}$, chosen by the adversary, and provide the lowest possible complexity.

- Choose $y \in \{0^{256}, 1^{256}\}$
- Preimage: $x = y$

Task 4: Birthday Paradox

c) Calculate the expected number of queries for finding a preimage $x \in \{0,1\}^{256}$ to a uniformly random sampled $y \twoheadleftarrow \{0,1\}^{256}$ for a success probability of 0.5.

Task 4: Birthday Paradox

c) Calculate the expected number of queries for finding a preimage $x \in \{0,1\}^{256}$ to a uniformly random sampled $y \twoheadleftarrow \{0,1\}^{256}$ for a success probability of 0.5.

We have 2^{256} possible hash values y, a random variable that y has hamming weight $\mathbf{Y} = i$, and need on average q_i attempts to find a preimage with probability 0.5 (this is a crude simplification):

$$\Pr[\mathbf{Y} = i] = \frac{\binom{n}{i}}{2^n}, \text{ and } q_i = \frac{\binom{n}{i}}{2}.$$

Then, we can define

$$\begin{aligned}\mathbb{E}[\mathbf{Y}] &= \sum_{i=0}^{n} \Pr[\mathbf{Y} = i] \cdot q_i = \sum_{i=0}^{n} \left(\frac{\binom{n}{i}}{2^n} \cdot \frac{\binom{n}{i}}{2} \right) \\ &= \frac{1}{2^{n+1}} \cdot \sum_{i=0}^{n} \binom{n}{i} \cdot \binom{n}{i} \approx 2^{250.17}.\end{aligned}$$

Questions?