

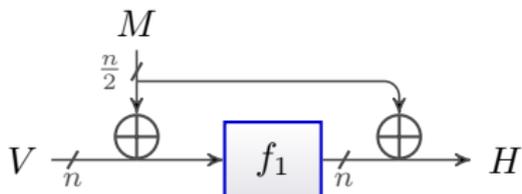
Thoughts on Permutation Based Hashing (Theoretical)

Martijn Stam

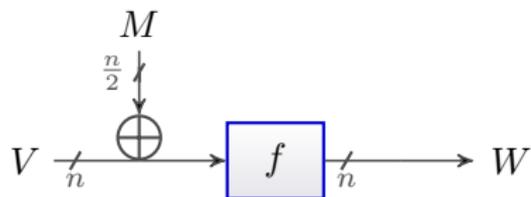
EPFL - LACAL

1st SHA-3 Candidate Conference

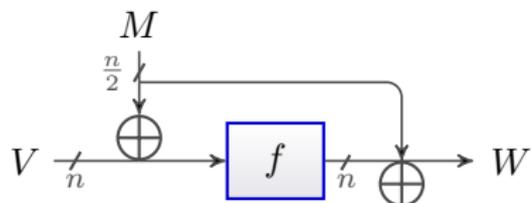
28 February 2009



Permutation Based Hashing

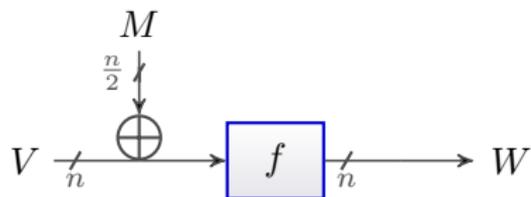


Sponge Construction



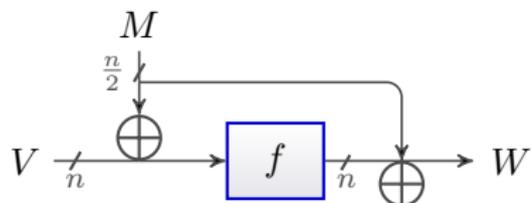
JH Construction

Permutation Based Hashing



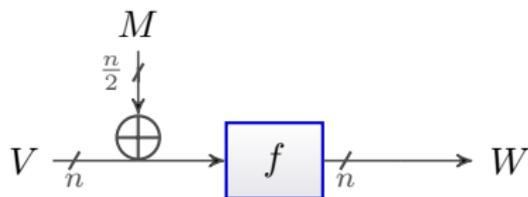
Sponge Construction

$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(q^2/2^{n/2})$$



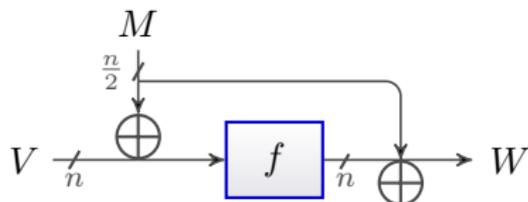
JH Construction

Permutation Based Hashing



Sponge Construction

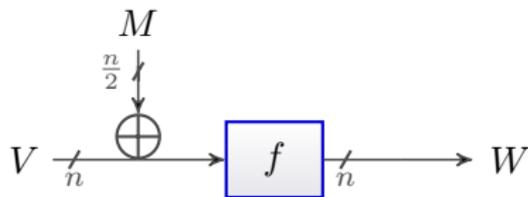
$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(q^2/2^{n/2})$$



JH Construction

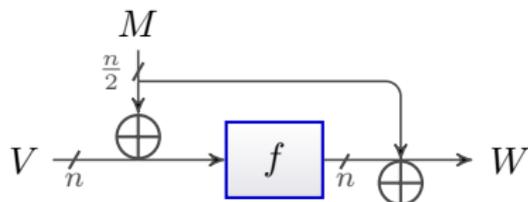
$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(n^d q^2 / 2^n)?$$

Permutation Based Hashing



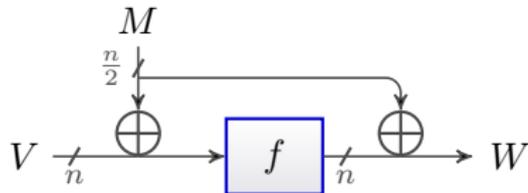
Sponge Construction

$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(q^2/2^{n/2})$$



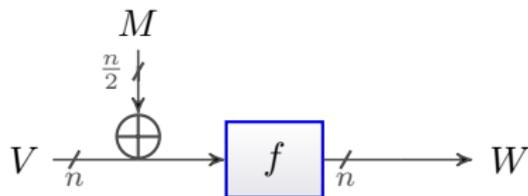
JH Construction

$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(n^d q^2/2^n)?$$



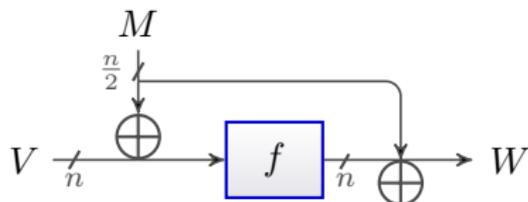
Variant Construction

Permutation Based Hashing



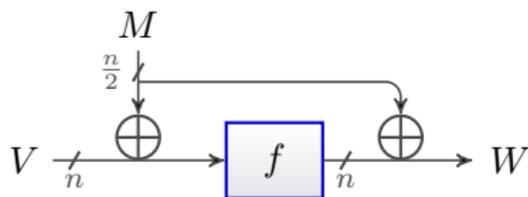
Sponge Construction

$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(q^2/2^{n/2})$$



JH Construction

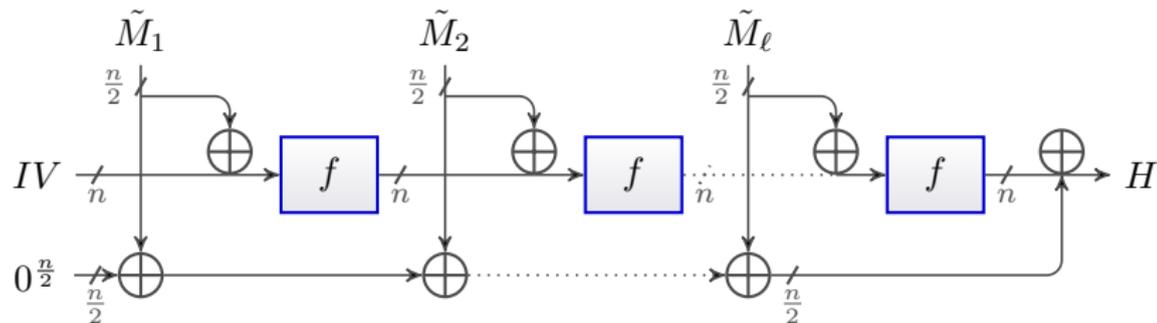
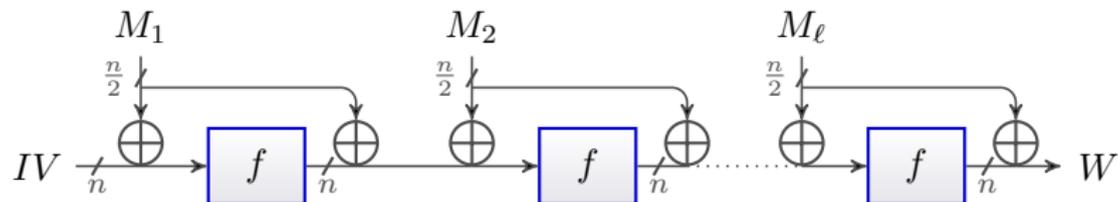
$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(n^d q^2/2^n)?$$



Variant Construction

$$\text{Adv}_{\mathcal{H}}^{\text{coll}}(q) = \Theta(q^2/n2^{n/2})$$

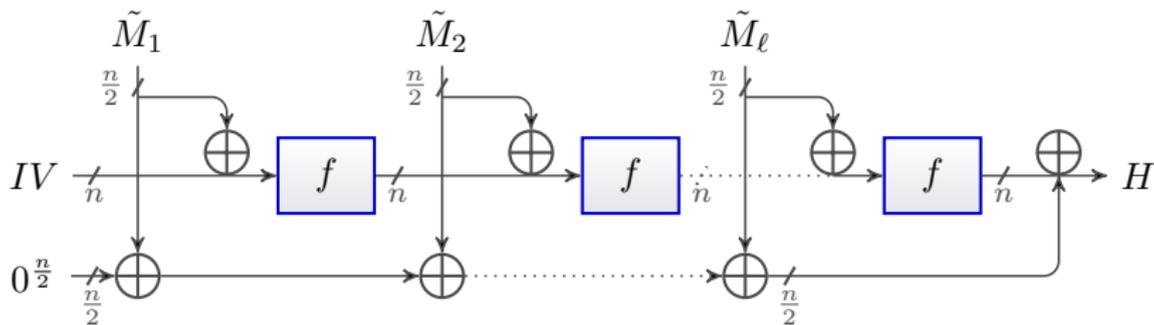
A Collision Attack on Bad JH



$$\tilde{M}_1 = M_1, \quad \tilde{M}_2 = M_1 + M_2, \quad \dots, \quad \tilde{M}_\ell = M_{\ell-1} + M_\ell$$

Finding the Collisions

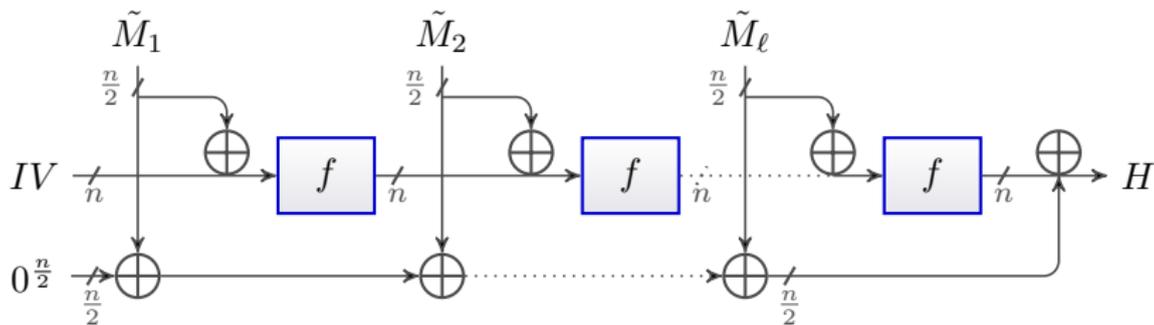
[GK08] Multicollision approach



- Find \tilde{M}_1^0 and \tilde{M}_1^1 that collide on the lower $n/2$ bits;
- Fix \tilde{M}_2^0 and \tilde{M}_2^1 to create a full collision on n bits.
- Let $Z_i = \tilde{M}_{2i-1}^0 + \tilde{M}_{2i-1}^1 + \tilde{M}_{2i}^0 + \tilde{M}_{2i}^1$ for $i = 1, \dots, n/2$.
- Find a combination such that $\sum_{i=1}^{n/2} Z_i^{b_i} = 0$.
- Set $\tilde{M}_i = \tilde{M}_i^0$ and $\tilde{M}'_i = \tilde{M}_i^{\left(b_{\lfloor \frac{i+1}{2} \rfloor}\right)}$,
Recover (M_1, \dots, M_ℓ) and (M'_1, \dots, M'_ℓ) .

Finding the Collisions

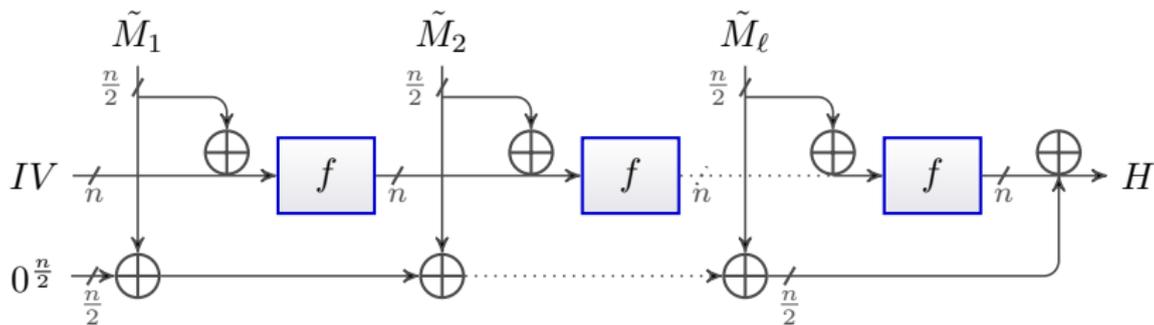
[GK08] Multicollision approach



- Find \tilde{M}_3^0 and \tilde{M}_3^1 that collide on the lower $n/2$ bits;
- ...
- Let $Z_i = \tilde{M}_{2i-1}^0 + \tilde{M}_{2i-1}^1 + \tilde{M}_{2i}^0 + \tilde{M}_{2i}^1$ for $i = 1, \dots, n/2$.
- Find a combination such that $\sum_{i=1}^{n/2} Z_i^{b_i} = 0$.
- Set $\tilde{M}_i = \tilde{M}_i^0$ and $\tilde{M}'_i = \tilde{M}_i^{\left(b_{\lfloor \frac{i+1}{2} \rfloor}\right)}$,
Recover (M_1, \dots, M_ℓ) and (M'_1, \dots, M'_ℓ) .

Finding the Collisions

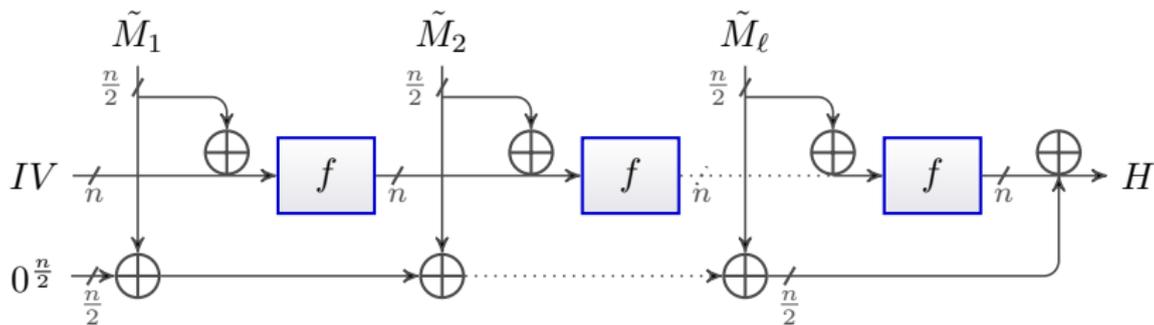
[GK08] Multicollision approach



- ...
- Fix \tilde{M}_n^0 and \tilde{M}_n^1 to create a full collision on n bits.
- Let $Z_i = \tilde{M}_{2i-1}^0 + \tilde{M}_{2i-1}^1 + \tilde{M}_{2i}^0 + \tilde{M}_{2i}^1$ for $i = 1, \dots, n/2$.
- Find a combination such that $\sum_{i=1}^{n/2} Z_i^{b_i} = 0$.
- Set $\tilde{M}_i = \tilde{M}_i^0$ and $\tilde{M}'_i = \tilde{M}_i^{\left(b_{\lfloor \frac{i+1}{2} \rfloor}\right)}$,
Recover (M_1, \dots, M_ℓ) and (M'_1, \dots, M'_ℓ) .

Finding the Collisions

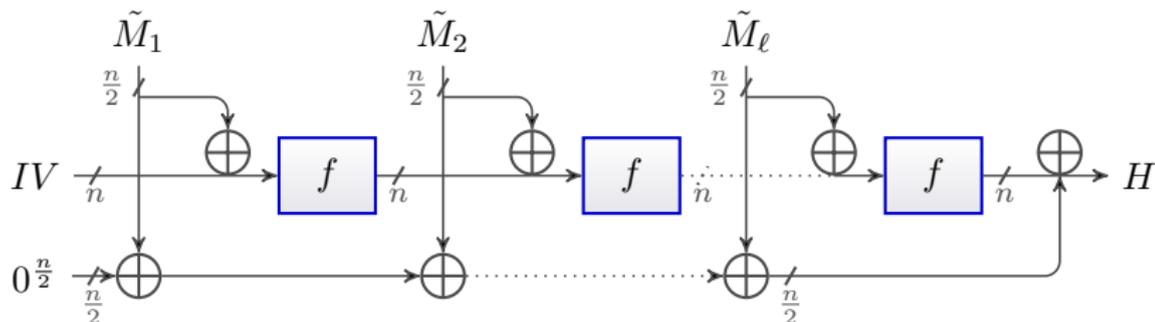
[GK08] Multicollision approach



- ...
- Fix \tilde{M}_n^0 and \tilde{M}_n^1 to create a full collision on n bits.
- Let $Z_i = \tilde{M}_{2i-1}^0 + \tilde{M}_{2i-1}^1 + \tilde{M}_{2i}^0 + \tilde{M}_{2i}^1$ for $i = 1, \dots, n/2$.
- Find a combination such that $\sum_{i=1}^{n/2} Z_i^{b_i} = 0$.
- Set $\tilde{M}_i = \tilde{M}_i^0$ and $\tilde{M}'_i = \tilde{M}_i^{\left(b_{\lfloor \frac{i+1}{2} \rfloor}\right)}$,
Recover (M_1, \dots, M_ℓ) and (M'_1, \dots, M'_ℓ) .

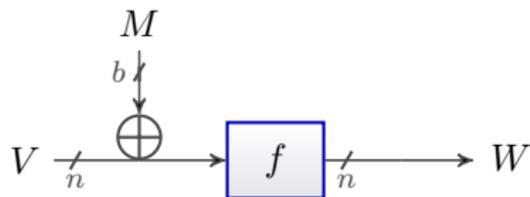
Finding the Collisions

[GK08] Multicollision approach

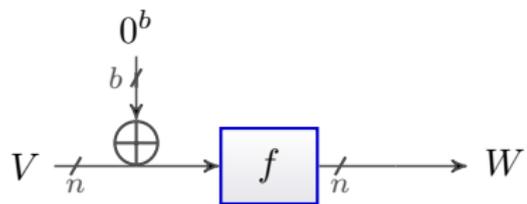


- ...
- Fix \tilde{M}_n^0 and \tilde{M}_n^1 to create a full collision on n bits.
- Let $Z_i = \tilde{M}_{2i-1}^0 + \tilde{M}_{2i-1}^1 + \tilde{M}_{2i}^0 + \tilde{M}_{2i}^1$ for $i = 1, \dots, n/2$.
- Find a combination such that $\sum_{i=1}^{n/2} Z_i^{b_i} = 0$.
- Set $\tilde{M}_i = \tilde{M}_i^0$ and $\tilde{M}'_i = \tilde{M}_i^{\left(b_{\lfloor \frac{i+1}{2} \rfloor}\right)}$,
Recover (M_1, \dots, M_ℓ) and (M'_1, \dots, M'_ℓ) .

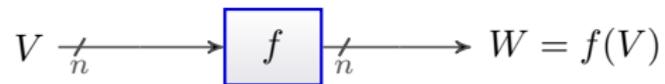
Basic Idea of Unreal Collisions



Basic Idea of Unreal Collisions

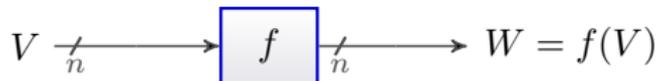


Basic Idea of Unreal Collisions

 0^b 

Basic Idea of Unreal Collisions

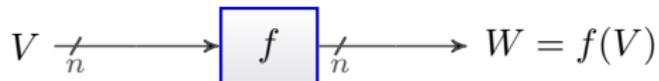
0^b



- $\forall V \mathcal{H}_V(0^b) = f(V) \Rightarrow \mathcal{H}_V((0^b)^m) = f^m(V)$.
- $f \in \text{Perm}(2^n)$ in a finite group of exponent $L = \text{lcm}(1, \dots, 2^n) | (2^n)!$
- Whatever f , f^L is the identify function.
- $\mathcal{H}_{IV}(0^{bL}1^b) = \mathcal{H}_{IV}(1^b) = \mathcal{H}_{IV}(1^b0^{bL})$

Basic Idea of Unreal Collisions

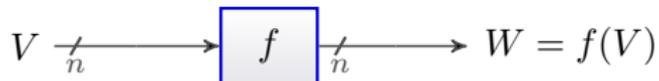
0^b



- $\forall V \mathcal{H}_V(0^b) = f(V) \Rightarrow \mathcal{H}_V((0^b)^m) = f^m(V)$.
- $f \in \text{Perm}(2^n)$ in a finite group of exponent $L = \text{lcm}(1, \dots, 2^n) | (2^n)!$
- Whatever f , f^L is the identify function.
- $\mathcal{H}_{IV}(0^{bL}1^b) = \mathcal{H}_{IV}(1^b) = \mathcal{H}_{IV}(1^b0^{bL})$

Basic Idea of Unreal Collisions

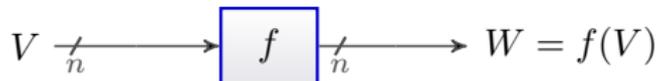
0^b



- $\forall V \mathcal{H}_V(0^b) = f(V) \Rightarrow \mathcal{H}_V((0^b)^m) = f^m(V)$.
- $f \in \text{Perm}(2^n)$ in a finite group of exponent $L = \text{lcm}(1, \dots, 2^n) | (2^n)!$
- Whatever f , f^L is the identify function.
- $\mathcal{H}_{IV}(0^{bL}1^b) = \mathcal{H}_{IV}(1^b) = \mathcal{H}_{IV}(1^b0^{bL})$

Basic Idea of Unreal Collisions

0^b



- $\forall V \mathcal{H}_V(0^b) = f(V) \Rightarrow \mathcal{H}_V((0^b)^m) = f^m(V)$.
- $f \in \text{Perm}(2^n)$ in a finite group of exponent $L = \text{lcm}(1, \dots, 2^n) | (2^n)!$
- Whatever f , f^L is the identity function.
- $\mathcal{H}_{IV}(0^{bL}1^b) = \mathcal{H}_{IV}(1^b) = \mathcal{H}_{IV}(1^b0^{bL})$

Conclusion

Collision in sponge-like construction essentially for free!!

No computation, no memory!

Consequences for Cubehash

- For Cubehash- r/b the permutation f consists of r rounds.
- Each round is the same permutation.
- Colliding message length is only bL/r .
- Reducing the number of rounds increases security!

Consequences for Cubehash

- For Cubehash- r/b the permutation f consists of r rounds.
- Each round is the same permutation.
- Colliding message length is only bL/r .
- Reducing the number of rounds increases security!

Optimized Parameter Set

We recommend Cubehash-1/128