Optimal Security for Keyed Hash Functions: Avoiding Time-Space Tradeoffs for Finding Collisions

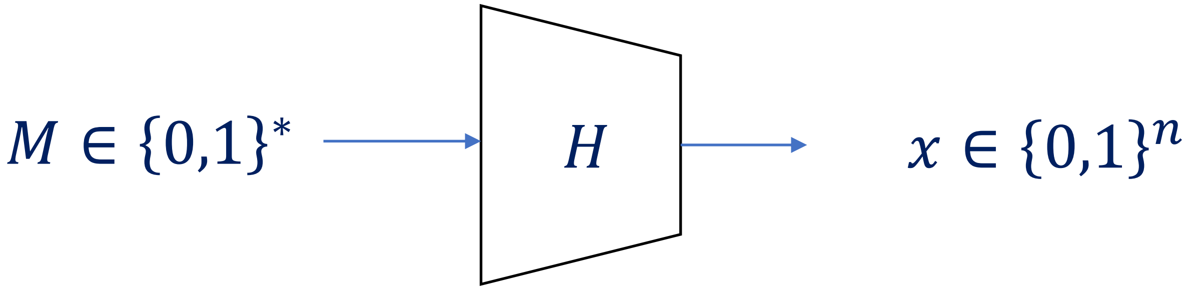
Cody Freitag
Cornell Tech

Ashrujit Ghoshal
University of Washington

Ilan Komargodski
Hebrew University and NTT Research

EUROCRYPT 2023

Cryptographic hash functions and collision resistance



Security properties:

- collision resistance
- one-wayness
- second pre-image resistance
- …

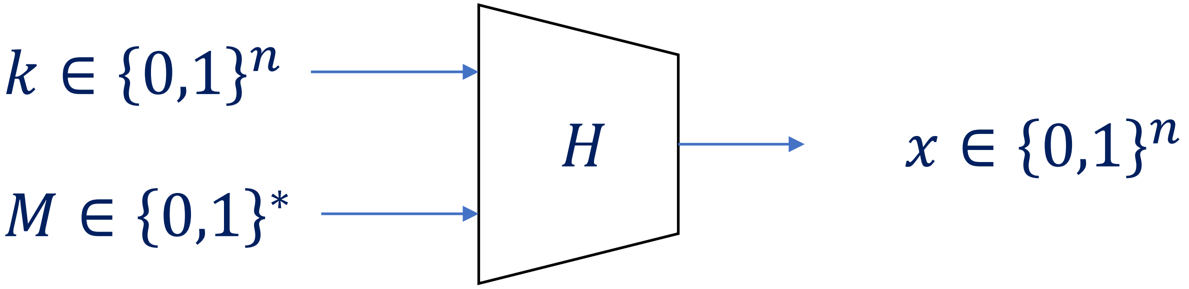
Applications:

- Hash and sign
- Proofs of Work
- Password authentication
- SNARKs
- …

Only relevant for uniform attackers
Non-uniform adversary can hardwire collisions

Keyed hash functions and collision resistance

Family of hash functions $\{H(k,.)\}_{k \in \{0,1\}^n}$



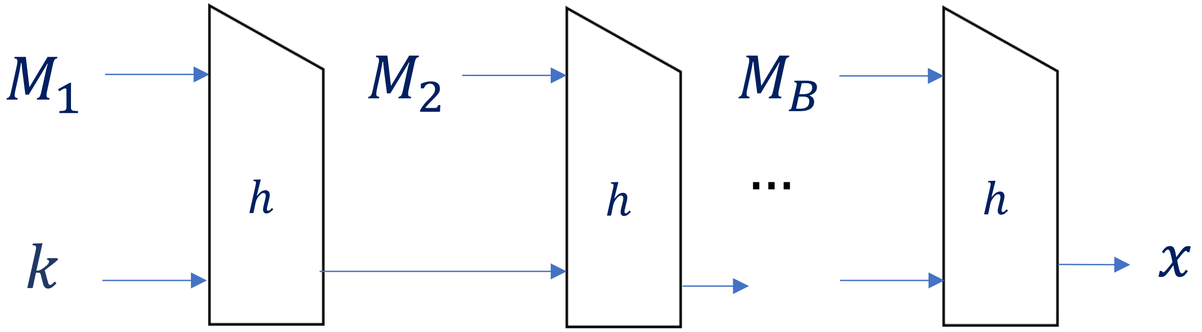
Collision resistance: For random k, hard to find $M \neq M'$: $H(k, M) = H(k, M')$

Given n, how would you build such H?

Practice for Building H

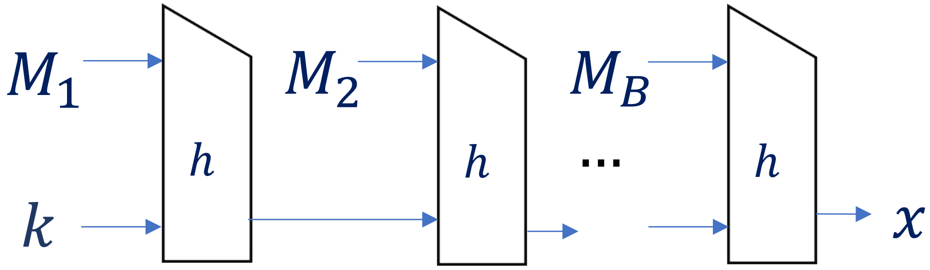
- Design a single $h: \{0,1\}^{2n} \to \{0,1\}^n$
- Iterate it in some way to get H: $\{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$

- (Keyed) Merkle-Damgård



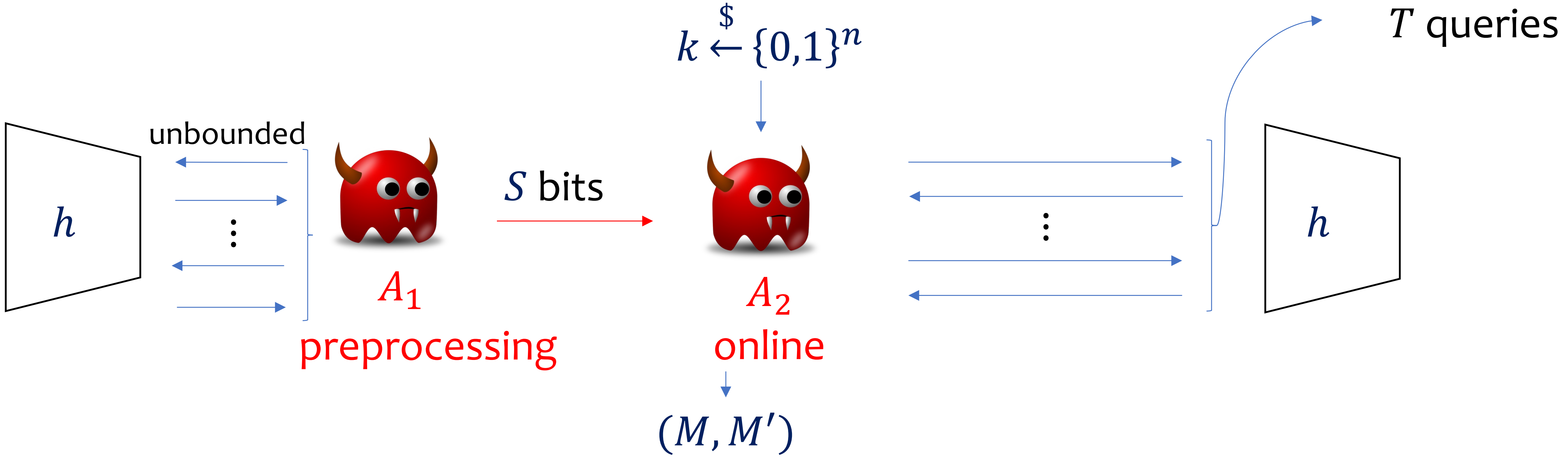
Back to Collision Resistance

- Is H collision resistant?
- Model $h: \{0,1\}^{2n} \rightarrow \{0,1\}^n$ as a random oracle
- Adversary is non-uniform



Auxilliary-Input Random Oracle Model (AI-ROM) [Unruh07]

$A = (A_1, A_2)$



$A = (A_1, A_2)$ wins if $M \neq M', H(k, M) = H(k, M')$

$$\mathrm{Adv}^{\mathrm{H}}(S,T) = \max_{(S,T)\text{ adv } A} \Pr[A \text{ wins}]$$

Establishing a baseline

A_1: Preprocessing

Remember collision for $\approx S$ different keys

A_2: Online

If key k not among the $\approx S$ keys, do birthday attack

$$\mathrm{Adv}^H(S,T) \geq \Omega\left(\frac{S}{2^n} + \frac{T^2}{2^n}\right)$$

Theorem. [DGK17]

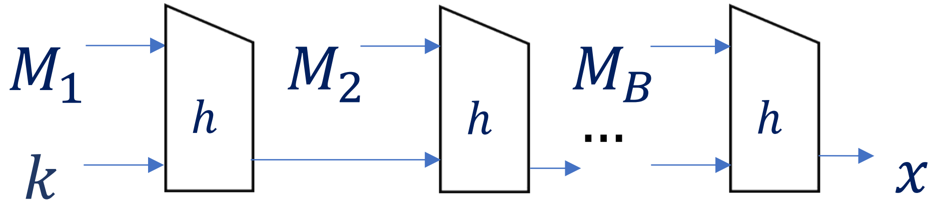
$$\mathrm{Adv}^G(S,T) \leq O\left(\frac{S}{2^n} + \frac{T^2}{2^n}\right)$$

Random $G: \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n$

What about keyed MD?

Time-space tradeoff for MD collisions

keyed MD



Theorem. [CDGS18] $\mathrm{Adv}^{\mathrm{MD}}(S,T) \geq \Omega\left(\frac{ST^2}{2^n}\right)$

Numerous follow up works analyzing various properties of keyed MD [ACDW20,GK22,AGL22]

Is this tradeoff inherent to any iterative construction?

What’s the right way of turning a single hash function into a keyed family of hash functions?

Is it possible to avoiding a security loss?

Our Results

Processing M bits

	Security	Assumption	# of h calls
MD	$ST^2/2^n$		M/n
H_1	$S/2^n + T^2/2^n$		M
H_2	$S/2^n + T^2/2^n$	$S < T$	$2M/n$
H_3	$S/2^n + T^2/2^n$	$ST^2 < 2^n$	$3M/n$

Follow from known results

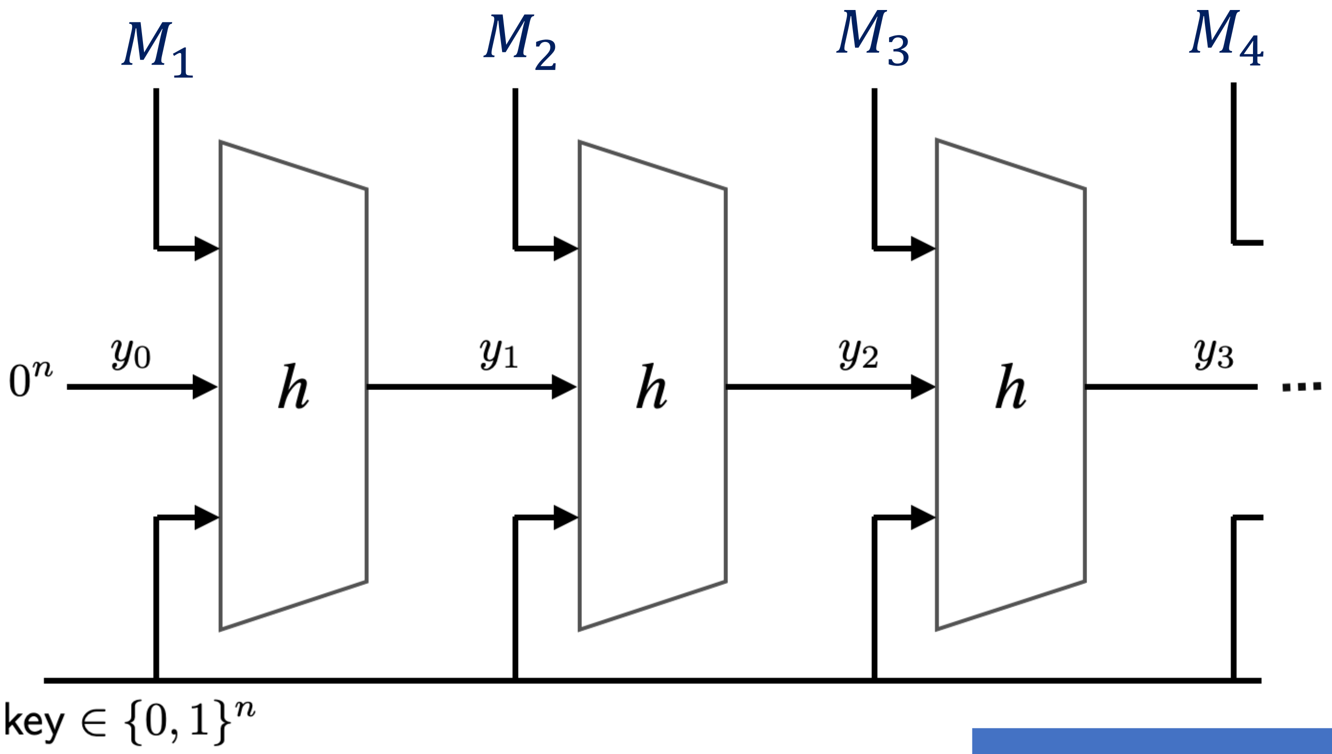
Hard & technical

Conjecture this is not needed

Construction H_1

[Goldwasser-Bellare 2008, uniform setting]

M = total input length

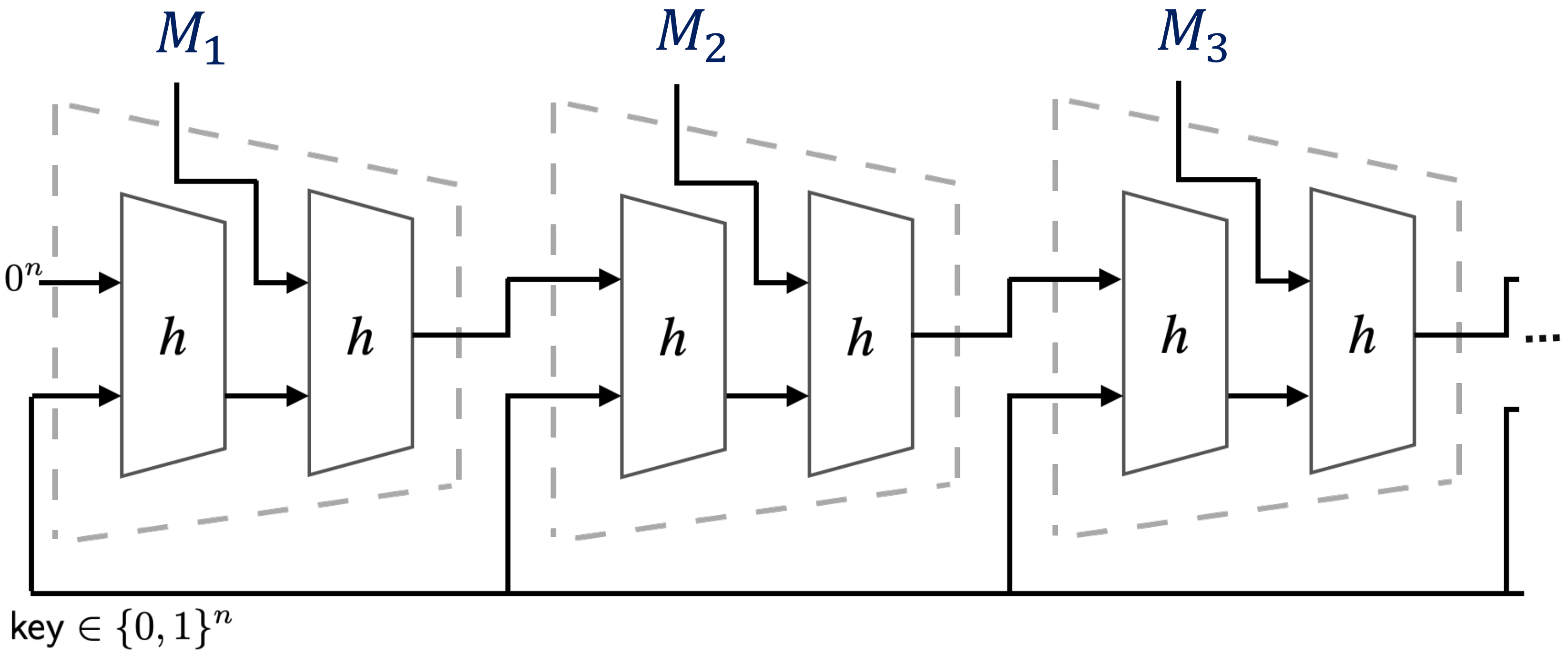


	Security	Assumption	# of h calls
H_1	$S/2^n + T^2/2^n$		M

By reduction to security of one-block case [DGK17]

Construction H_2

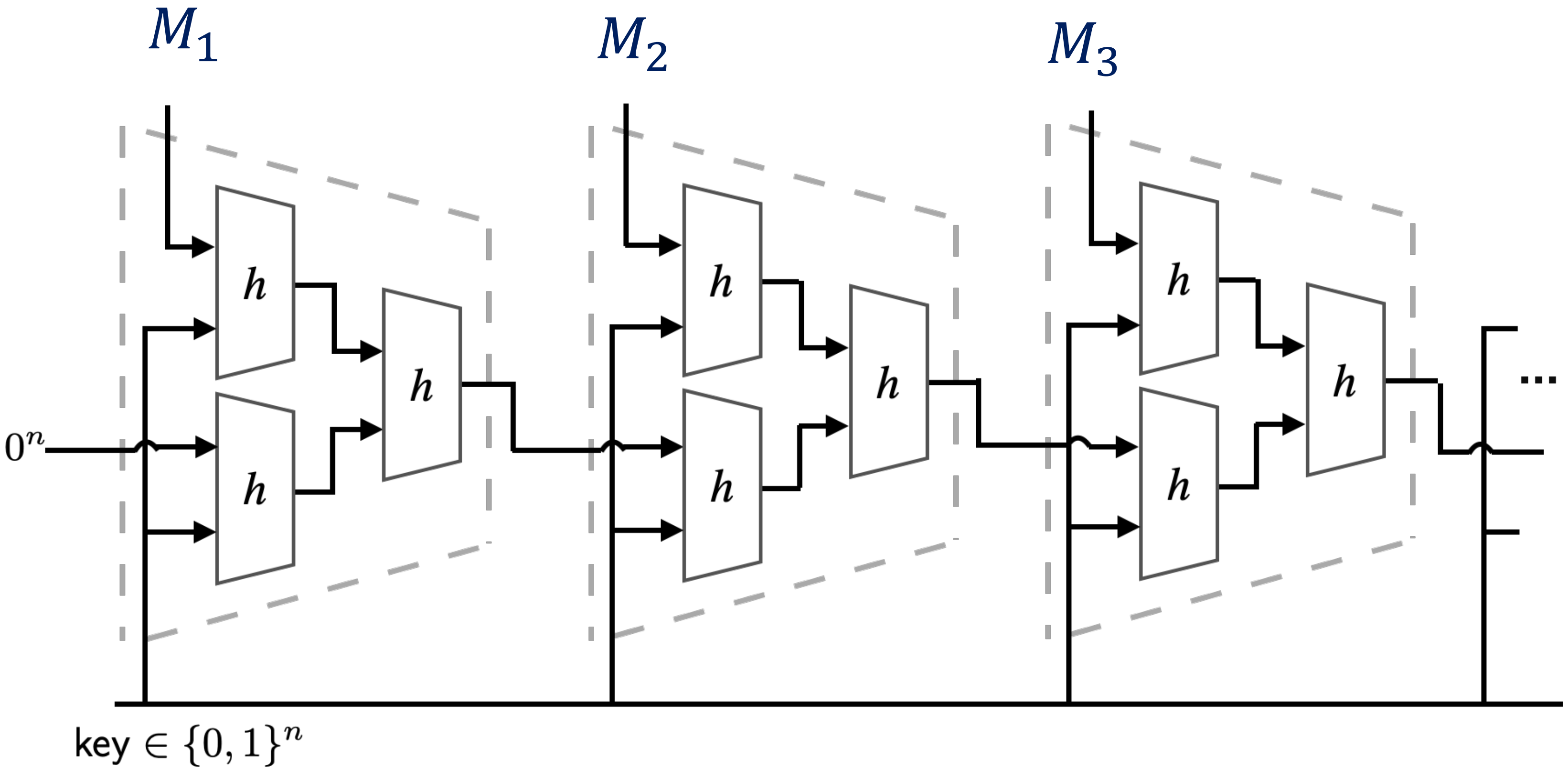
M = total input length



	Security	Assumption	# of h calls
H_2	$S/2^n + T^2/2^n$	$S < T$	$2M/n$

By reduction to security of two-block case [ACDW20]

Construction H_3



M = total input length

	Security	Assumption	# of h calls
H_3	$S/2^n + T^2/2^n$	$ST^2 < 2^n$	$3M/n$

Proof via the multi-instance framework [AGL22]

Conclusions

- New way of building keyed families of hash function
  - Via Merkle-tree-based keyed hashing approach
- Prior works focus on analyzing existing weak variants

Open problems

- Prove conjectured security of H_3 for $ST^2 > 2^n$
- Other preprocessing resistant constructions

Paper: ePrint/2023/348