Radbound University

Multimixer-128: Universal Keyed Hashing Based on Integer Multiplication

Koustabh Ghosh, Parisa Amiri Eliasi, Joan Daemen

Radboud University, Nijmegen, the Netherlands

FSE presentation

March 25, 2024

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build
  - Message authentication code (mac) functions [WC81]

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build
  - Message authentication code (mac) functions [WC81]
  - Doubly-extendable cryptographic keyed (deck) functions [Dae+18]

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build
  - Message authentication code (mac) functions [WC81]
  - Doubly-extendable cryptographic keyed (deck) functions [Dae+18]
- The security of a keyed hash function $F_{\mathbf{K}}$ is determined by its universality [Sti95]:

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build
  - Message authentication code (mac) functions [WC81]
  - Doubly-extendable cryptographic keyed (deck) functions [Dae+18]
- The security of a keyed hash function $F_{\mathbf{K}}$ is determined by its universality [Sti95]:
  - $F_{\mathbf{K}}$ is ε-universal $\implies \forall\, \mathbf{M} \neq \mathbf{M}^*,\ \Pr[F_{\mathbf{K}}(\mathbf{M}) = F_{\mathbf{K}}(\mathbf{M}^*)] \leq \varepsilon$

Keyed hash functions

- Keyed hash functions are a class of cryptographic primitives that
- Compress variable-length inputs to a fixed sized state under a secret key
- Keyed hash functions can be used to build
  - Message authentication code (mac) functions [WC81]
  - Doubly-extendable cryptographic keyed (deck) functions [Dae+18]
- The security of a keyed hash function $F_{\mathbf{K}}$ is determined by its universality [Sti95]:
  - $F_{\mathbf{K}}$ is ε-universal $\implies \forall\, \mathbf{M} \neq \mathbf{M}^*,\ \Pr[F_{\mathbf{K}}(\mathbf{M}) = F_{\mathbf{K}}(\mathbf{M}^*)] \leq \varepsilon$
  - $F_{\mathbf{K}}$ is ε-Δuniversal $\implies \forall\, \mathbf{M} \neq \mathbf{M}^*,\ \Pr[F_{\mathbf{K}}(\mathbf{M}) - F_{\mathbf{K}}(\mathbf{M}^*) = \Delta] \leq \varepsilon$

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f\colon G \to G'$ ...

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f\colon G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}[f]$ can be built with

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f \colon G \to G'$ ...
- A keyed hash function $F = \text{Parallel}[f]$ can be built with
  - The key space is G^{κ}

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f: G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}[f]$ can be built with
  - The key space is G^κ
  - The message space is $\bigcup_{\ell=1}^{\kappa} G^\ell$

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f\colon G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}[f]$ can be built with
  - The key space is G^κ
  - The message space is $\bigcup_{\ell=1}^{\kappa} G^\ell$
  - The digest space is G'

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f\colon G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}\,[f]$ can be built with
  - The key space is G^κ
  - The message space is $\bigcup_{\ell=1}^{\kappa} G^\ell$
  - The digest space is G'
- For a message $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ and key $\mathbf{K} = (K_0, K_1 \ldots, K_{\kappa-1})$,

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f : G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}[f]$ can be built with
  - The key space is G^κ
  - The message space is $\bigcup_{\ell=1}^{\kappa} G^\ell$
  - The digest space is G'
- For a message $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ and key $\mathbf{K} = (K_0, K_1 \ldots, K_{\kappa-1})$,
  - $\mathbf{M} + \mathbf{K} = (M_0 + K_0, M_1 + K_1, \ldots, M_{|\mathbf{M}|-1} + K_{|\mathbf{M}|-1})$

The parallel construction

- Our goal: Design ε-Δuniversal keyed hash function with $\varepsilon \approx 2^{-128}$
- That are efficient for software platforms
- Our design strategy: Parallelization of a public function [Gho+23],
- Which is the generalization of the parallelization of a public permutations [FRD23]
- From a public function $f\colon G \to G'$ ...
- A keyed hash function $F = \mathsf{Parallel}[f]$ can be built with
  - The key space is G^κ
  - The message space is $\bigcup_{\ell=1}^{\kappa} G^\ell$
  - The digest space is G'
- For a message $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ and key $\mathbf{K} = (K_0, K_1 \ldots, K_{\kappa-1})$,
  - $\mathbf{M} + \mathbf{K} = (M_0 + K_0, M_1 + K_1, \ldots, M_{|\mathbf{M}|-1} + K_{|\mathbf{M}|-1})$
  - $F_{\mathbf{K}}(M) := F(\mathbf{M} + \mathbf{K})$

Parallel $[f]$

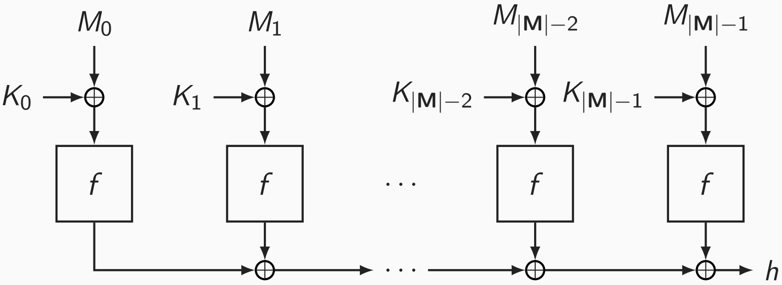

Figure: The parallelization of f: Parallel $[f]$ [FRD23]

Universality of Parallel $[f]$

- For the fixed length public function f . . .

Universality of Parallel $[f]$

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$

Universality of Parallel $[f]$

- For the fixed length public function f ...
    - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$
    - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G | f(X) = Z\}}{\#G}$

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A,\Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G | f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G | f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A, \Delta)$

Universality of Parallel $[f]$

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G | f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A, \Delta)$
    - $\mathrm{MIP}_f = \max_{Z} \mathrm{IP}_f(Z)$

Universality of Parallel $[f]$

- For the fixed length public function $f\ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A,\Delta) = \frac{\#\{X \in G \mid f(X+A)-f(X)=\Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G \mid f(X)=Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A,\Delta)$
    - $\mathrm{MIP}_f = \max_{Z} \mathrm{IP}_f(Z)$
- Parallel $[f]$ is $\max\{\mathrm{MDP}_f, \mathrm{MIP}_f\}$-$\Delta$universal [Gho+23]

Universality of Parallel $[f]$

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G | f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G | f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A, \Delta)$
    - $\mathrm{MIP}_f = \max_{Z} \mathrm{IP}_f(Z)$
- Parallel $[f]$ is $\max\{\mathrm{MDP}_f, \mathrm{MIP}_f\}$-$\Delta$universal [Gho+23]
- Obtaining universality of Parallel $[f]$ is reduced to obtaining MDP_f and MIP_f

- For the fixed length public function f ...
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G \mid f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G \mid f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A, \Delta)$
    - $\mathrm{MIP}_f = \max_{Z} \mathrm{IP}_f(Z)$
- Parallel $[f]$ is $\max\{\mathrm{MDP}_f, \mathrm{MIP}_f\}$-$\Delta$universal [Gho+23]
- Obtaining universality of Parallel $[f]$ is reduced to obtaining MDP_f and MIP_f
- Universality not only takes into account messages of equal length,

Universality of Parallel $[f]$

- For the fixed length public function $f \ldots$
  - The DP of a differential (A, Δ) is: $\mathrm{DP}_f(A, \Delta) = \frac{\#\{X \in G \mid f(X+A) - f(X) = \Delta\}}{\#G}$
  - The IP of any output Z of f is: $\mathrm{IP}_f(Z) = \frac{\#\{X \in G \mid f(X) = Z\}}{\#G}$
  - The maximum possible value of DP_f and IP_f are denoted as:
    - $\mathrm{MDP}_f = \max_{A \neq 0, \Delta} \mathrm{DP}_f(A, \Delta)$
    - $\mathrm{MIP}_f = \max_Z \mathrm{IP}_f(Z)$
- Parallel $[f]$ is $\max\{\mathrm{MDP}_f, \mathrm{MIP}_f\}$-$\Delta$universal [Gho+23]
- Obtaining universality of Parallel $[f]$ is reduced to obtaining MDP_f and MIP_f
- Universality not only takes into account messages of equal length,
- But also messages of variable length

NH$[\kappa, w]$ [Bla+99]

- **NH**$[\kappa, w]$: very fast keyed hash function on software with

NH$[\kappa, w]$ [Bla+99]

- **NH**$[\kappa, w]$: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)

NH[κ, w] [Bla+99]

- **NH**[κ, w]: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)
- Performance is due to fast integer multiplication instructions

$\mathbf{NH}[\kappa, w]$ [Bla+99]

- $\mathbf{NH}[\kappa, w]$: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)
- Performance is due to fast integer multiplication instructions
- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ can be viewed as the parallelization of

$$M[w]\colon (\mathbb{Z}/2^w\mathbb{Z})^2 \to \mathbb{Z}/2^{2w}\mathbb{Z}\colon (x, y) \mapsto x \times y$$

- $\mathbf{NH}[\kappa, w]$: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)
- Performance is due to fast integer multiplication instructions
- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ can be viewed as the parallelization of

$$M[w]\colon (\mathbb{Z}/2^w\mathbb{Z})^2 \to \mathbb{Z}/2^{2w}\mathbb{Z}\colon (x, y) \mapsto x \times y$$

- For $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ with $M_i = (m_{2i}, m_{2i+1}) \in (\mathbb{Z}/2^w\mathbb{Z})^2$,

- $\mathbf{NH}[\kappa, w]$: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)
- Performance is due to fast integer multiplication instructions
- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ can be viewed as the parallelization of

$$M[w]\colon (\mathbb{Z}/2^w\mathbb{Z})^2 \to \mathbb{Z}/2^{2w}\mathbb{Z}\colon (x, y) \mapsto x \times y$$

- For $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ with $M_i = (m_{2i}, m_{2i+1}) \in (\mathbb{Z}/2^w\mathbb{Z})^2$,
- And $\mathbf{K} = (K_0, K_1 \ldots, K_{\kappa/2-1})$ with $K_i = (k_{2i}, k_{2i+1}) \in (\mathbb{Z}/2^w\mathbb{Z})^2$

NH$[\kappa, w]$ [Bla+99]

- **NH**$[\kappa, w]$: very fast keyed hash function on software with
- An even $\kappa \geq 2$ (blocksize) and $w \geq 1$ (wordsize)
- Performance is due to fast integer multiplication instructions
- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ can be viewed as the parallelization of

$$M[w]\colon (\mathbb{Z}/2^w\mathbb{Z})^2 \to \mathbb{Z}/2^{2w}\mathbb{Z}\colon (x, y) \mapsto x \times y$$

- For $\mathbf{M} = (M_0, M_1, \ldots, M_{|\mathbf{M}|-1})$ with $M_i = (m_{2i}, m_{2i+1}) \in (\mathbb{Z}/2^w\mathbb{Z})^2$,
- And $\mathbf{K} = (K_0, K_1 \ldots, K_{\kappa/2-1})$ with $K_i = (k_{2i}, k_{2i+1}) \in (\mathbb{Z}/2^w\mathbb{Z})^2$

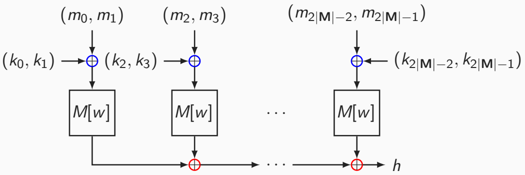

Figure: $\mathbf{NH}_{\mathbf{K}}[\kappa, w] = \text{Parallel}\,[M[w]]$

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages

Extending universality of NH

- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH}_{\mathbf{K}_0}[\kappa, w](M) \mid\mid \ldots \mid\mid \mathbf{NH}_{\mathbf{K}_{t-1}}[\kappa, w](M)$ is 2^{-wt}-universal

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH_{K_0}}[\kappa, w](M) \,||\, \ldots \,||\, \mathbf{NH_{K_{t-1}}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH_{K_0}}[\kappa, w](M) \,||\, \ldots \,||\, \mathbf{NH_{K_{t-1}}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH^T_K}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...

Extending universality of NH

- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH}_{\mathbf{K}_0}[\kappa, w](M) \mid\mid \ldots \mid\mid \mathbf{NH}_{\mathbf{K}_{t-1}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material

Extending universality of NH

- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH}_{\mathbf{K}_0}[\kappa, w](M) \parallel \ldots \parallel \mathbf{NH}_{\mathbf{K}_{t-1}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material
- So, $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ is 2^{-128}-universal requiring only 192-bits of extra key material

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH_{K_0}}[\kappa, w](M) \parallel \ldots \parallel \mathbf{NH_{K_{t-1}}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH_K^T}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material
- So, $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-128}-universal requiring only 192-bits of extra key material
- And is the fastest option for keyed hashing on our target platforms

Extending universality of NH

- $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH}_{\mathbf{K}_0}[\kappa, w](M) \mathbin{||} \ldots \mathbin{||} \mathbf{NH}_{\mathbf{K}_{t-1}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material
- So, $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ is 2^{-128}-universal requiring only 192-bits of extra key material
- And is the fastest option for keyed hashing on our target platforms
- A mode is defined on top of $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ to

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH}_{\mathbf{K}_0}[\kappa, w](M) \parallel \ldots \parallel \mathbf{NH}_{\mathbf{K}_{t-1}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material
- So, $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ is 2^{-128}-universal requiring only 192-bits of extra key material
- And is the fastest option for keyed hashing on our target platforms
- A mode is defined on top of $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ to
  - Handle messages of arbitrary length

Extending universality of NH

- $\mathbf{NH_K}[\kappa, w]$ is 2^{-w}-universal on equal-length messages
- $\mathbf{NH_{K_0}}[\kappa, w](M) \mid\mid \ldots \mid\mid \mathbf{NH_{K_{t-1}}}[\kappa, w](M)$ is 2^{-wt}-universal
- But, this requires a $t\kappa w$-bit key
- $\mathbf{NH_K^T}[\kappa, w, t]$ is 2^{-wt}-universal on equal-length messages, and ...
- Only requires $2w(t-1)$-bits of extra key material
- So, $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-128}-universal requiring only 192-bits of extra key material
- And is the fastest option for keyed hashing on our target platforms
- A mode is defined on top of $\mathbf{NH_K^T}[\kappa, 32, 4]$ to
  - Handle messages of arbitrary length
  - Ensure that universality bound holds in case of messages of variable length

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework

Differential properties of M$[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$

- We analyze security of $\mathbf{NH}_{\mathbf{K}}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages
- Our approach also leads to

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages
- Our approach also leads to
  - A tight upper-bound for $\max_\delta \mathrm{DP}_{M[w]}((a, b), \delta)$

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages
- Our approach also leads to
  - A tight upper-bound for $\max_\delta \mathrm{DP}_{M[w]}((a, b), \delta)$
  - The value of $\mathrm{DP}_{M[w]}((a, b), 0)$

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages
- Our approach also leads to
  - A tight upper-bound for $\max_\delta \mathrm{DP}_{M[w]}((a, b), \delta)$
  - The value of $\mathrm{DP}_{M[w]}((a, b), 0)$
- $\mathrm{DP}_{M[w]}$ is upper-bounded by 2^{-w}, but ...

Differential properties of $\mathsf{M}[w]$

- We analyze security of $\mathbf{NH_K}[\kappa, w]$ in the parallelized public function framework
- This reduces the problem to obtaining the values of $\mathrm{MIP}_{M[w]}$ and $\mathrm{MDP}_{M[w]}$
- We obtain $\mathrm{MIP}_{M[w]} \leq 2^{-w+1}$ and $\mathrm{MDP}_{M[w]} = 2^{-w}$
- This means that $\mathbf{NH_K^T}[\kappa, 32, 4]$ is 2^{-124}-ΔUniversal over all messages
- Our approach also leads to
  - A tight upper-bound for $\max_\delta \mathrm{DP}_{M[w]}((a, b), \delta)$
  - The value of $\mathrm{DP}_{M[w]}((a, b), 0)$
- $\mathrm{DP}_{M[w]}$ is upper-bounded by 2^{-w}, but ...
- Only for input differences of the type $(a, 0), (0, a), (a, a), (a, -a)$

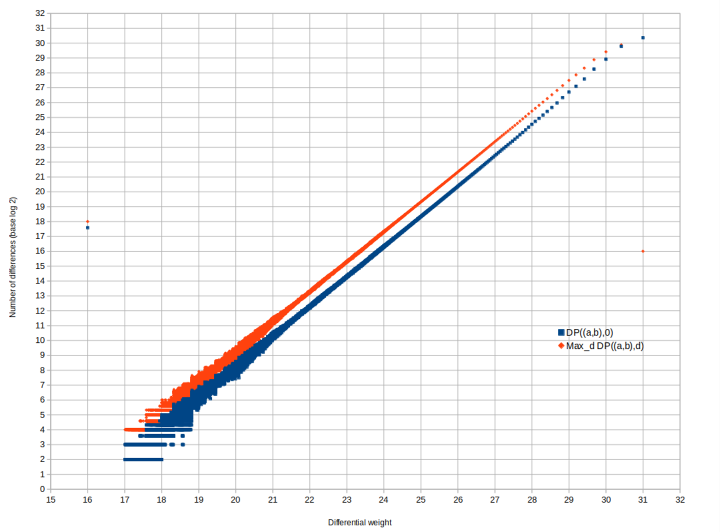

Figure: Upper-bound of $\max_\delta \mathrm{DP}_{M[16]}((a,b),\delta)$, $\mathrm{DP}_{M[16]}((a,b),0)$ vs. Number of differences

$\mathcal{F}$-128, the public function of Multimixer-128

- $M[w]$ is an excellent choice due to the propagation properties

- $M[w]$ is an excellent choice due to the propagation properties
- We present the public function $\mathcal{F}$-128

$\mathcal{F}$-128, the public function of Multimixer-128

- $M[w]$ is an excellent choice due to the propagation properties
- We present the public function $\mathcal{F}$-128
- Multimixer-128 is simply Parallel [$\mathcal{F}$-128]

$\mathcal{F}$-128, the public function of Multimixer-128

- $M[w]$ is an excellent choice due to the propagation properties
- We present the public function $\mathcal{F}$-128
- Multimixer-128 is simply Parallel $[\mathcal{F}$-128$]$
- $\mathcal{F}$-128: $\left(\mathbb{Z}/2^{32}\mathbb{Z}\right)^4 \times \left(\mathbb{Z}/2^{32}\mathbb{Z}\right)^4 \rightarrow \left(\mathbb{Z}/2^{64}\mathbb{Z}\right)^8$ is defined as

$\mathcal{F}$-128, the public function of Multimixer-128

- $M[w]$ is an excellent choice due to the propagation properties
- We present the public function $\mathcal{F}$-128
- Multimixer-128 is simply Parallel [$\mathcal{F}$-128]
- $\mathcal{F}\text{-}128\colon (\mathbb{Z}/2^{32}\mathbb{Z})^4 \times (\mathbb{Z}/2^{32}\mathbb{Z})^4 \to (\mathbb{Z}/2^{64}\mathbb{Z})^8$ is defined as
- $\mathcal{F}\text{-}128(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \odot \mathbf{y}, \mathrm{N}_\alpha \cdot \mathbf{x} \odot \mathrm{N}_\beta \cdot \mathbf{y})$

- $M[w]$ is an excellent choice due to the propagation properties
- We present the public function $\mathcal{F}$-128
- Multimixer-128 is simply Parallel [$\mathcal{F}$-128]
- $\mathcal{F}\text{-}128 \colon (\mathbb{Z}/2^{32}\mathbb{Z})^4 \times (\mathbb{Z}/2^{32}\mathbb{Z})^4 \to (\mathbb{Z}/2^{64}\mathbb{Z})^8$ is defined as
- $\mathcal{F}\text{-}128(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \odot \mathbf{y}, \mathrm{N}_\alpha \cdot \mathbf{x} \odot \mathrm{N}_\beta \cdot \mathbf{y})$

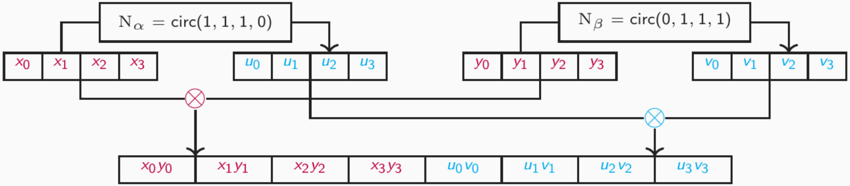

Figure: $\mathcal{F}$-128, the public function of Multimixer-128

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x} \neq \mathbf{0}}(\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N} \cdot \mathbf{x}))$

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x} \neq \mathbf{0}} (\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N} \cdot \mathbf{x}))$
- N_α and N_β both have branch number 4

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x} \neq \mathbf{0}}(\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N} \cdot \mathbf{x}))$
- N_α and N_β both have branch number 4
- When $(\mathbf{a}, \mathbf{b})$ is such that 4 multiplications are active, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{b}), \Delta) \leq 2^{-128}$

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x} \neq \mathbf{0}}(\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N} \cdot \mathbf{x}))$
- N_α and N_β both have branch number 4
- When $(\mathbf{a}, \mathbf{b})$ is such that 4 multiplications are active, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{b}), \Delta) \leq 2^{-128}$
- In particular for all $\mathbf{a} \neq \mathbf{0}$, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{0}), \mathbf{0}) = \mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{0}, \mathbf{a}), \mathbf{0}) = 2^{-128}$

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x} \neq \mathbf{0}}(\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N} \cdot \mathbf{x}))$
- N_α and N_β both have branch number 4
- When $(\mathbf{a}, \mathbf{b})$ is such that 4 multiplications are active, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{b}), \Delta) \leq 2^{-128}$
- In particular for all $\mathbf{a} \neq \mathbf{0}$, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{0}), \mathbf{0}) = \mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{0}, \mathbf{a}), \mathbf{0}) = 2^{-128}$
- For all other differences, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a}, \mathbf{b}), \Delta) \leq 2^{-160}$

Universality of Multimixer-128

- We prove for $\mathbf{Z} \neq \mathbf{0}$, $\mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{Z}) \ll \mathrm{IP}_{\mathcal{F}\text{-}128}(\mathbf{0}) = \frac{2^{129}-1}{2^{256}} \leq 2^{-127} = \mathrm{MIP}_{\mathcal{F}\text{-}128}$
- $\mathrm{MDP}_{\mathcal{F}\text{-}128}$ is determined by the number of minimum active multiplications
- The minimum number of active multiplications is determined by N_α and N_β
- Branch number of an $n \times n$ matrix N over $\mathbb{Z}/2^w\mathbb{Z}$: $\min_{\mathbf{x}\neq\mathbf{0}}(\mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathrm{N}\cdot\mathbf{x}))$
- N_α and N_β both have branch number 4
- When $(\mathbf{a},\mathbf{b})$ is such that 4 multiplications are active, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a},\mathbf{b}),\Delta) \leq 2^{-128}$
- In particular for all $\mathbf{a} \neq \mathbf{0}$, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a},\mathbf{0}),\mathbf{0}) = \mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{0},\mathbf{a}),\mathbf{0}) = 2^{-128}$
- For all other differences, $\mathrm{DP}_{\mathcal{F}\text{-}128}((\mathbf{a},\mathbf{b}),\Delta) \leq 2^{-160}$
- Thus, Multimixer-128 is ε-Δ universal with

$$\varepsilon = \max\{\mathrm{MDP}_{\mathcal{F}\text{-}128}, \mathrm{MIP}_{\mathcal{F}\text{-}128}\} = 2^{-127}$$

Algorithm	# ops.\ per 256-bit input		
	$\times$	$+$ mod 2^{32}	$+$ mod 2^{64}
$\mathbf{NH}_{\mathbf{K}}^{\mathsf{T}}[\kappa, 32, 4]$	16	32	16
Multimixer-128	8	20	8

Table: Comparison of # arithmetic operations

Implementation and Benchmarking Results

Algorithm	# ops.\ per 256-bit input		
	$\times$	$+$ mod 2^{32}	$+$ mod 2^{64}
$\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$	16	32	16
Multimixer-128	8	20	8

Table: Comparison of # arithmetic operations

Algorithm	# Instructions\ per 256-bit input	Input length in bytes		
		512	4096	32768
$\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$	16	2.033	1.500	1.558
Multimixer-128	11	1.830	1.233	1.396

Table: Performance on 32-bit ARMv7 Cortex-A processor in cycles per byte

Implementation and Benchmarking Results

Algorithm	# ops.\ per 256-bit input		
	$\times$	$+$ mod 2^{32}	$+$ mod 2^{64}
$\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$	16	32	16
Multimixer-128	8	20	8

Table: Comparison of # arithmetic operations

Algorithm	# Instructions\ per 256-bit input	Input length in bytes		
		512	4096	32768
$\mathbf{NH}^{\mathsf{T}}_{\mathbf{K}}[\kappa, 32, 4]$	16	2.033	1.500	1.558
Multimixer-128	11	1.830	1.233	1.396

Table: Performance on 32-bit ARMv7 Cortex-A processor in cycles per byte

Thank you for your attention!

References

[WC81] Mark N. Wegman and Larry Carter. **“New Hash Functions and Their Use in Authentication and Set Equality”.** In: *J. Comput. Syst. Sci.* 22.3 (1981), pp. 265–279.

[Sti95] Douglas R. Stinson. **“On the Connections Between Universal Hashing, Combinatorial Designs and Error-Correcting Codes”.** In: *Electron. Colloquium Comput. Complex.* TR95-052 (1995). ECCC: TR95-052.

[Bla+99] John Black et al. **“UMAC: Fast and Secure Message Authentication”.** In: *Advances in Cryptology - CRYPTO ’99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Ed. by Michael J. Wiener. Vol. 1666. LNCS. Springer, 1999, pp. 216–233.

[Dae+18] Joan Daemen et al. **“The design of Xoodoo and Xoofff”.** In: *IACR Trans. Symmetric Cryptol.* 2018.4 (2018), pp. 1–38.

[FRD23] Jonathan Fuchs, Yann Rotella, and Joan Daemen. **“On the Security of Keyed Hashing Based on Public Permutations”.** In: *Advances in*

Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III. Vol. 14083. Lecture Notes in Computer Science. Springer, 2023, pp. 607–627.

[Gho+23] Koustabh Ghosh et al. **"Universal Hashing Based on Field Multiplication and (Near-)MDS Matrices".** In: *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*. Vol. 14064. Lecture Notes in Computer Science. Springer, 2023, pp. 129–150.