Radbound University

Permutation-Based Hashing Beyond the Birthday Bound

Charlotte Lefevre, Bart Mennink

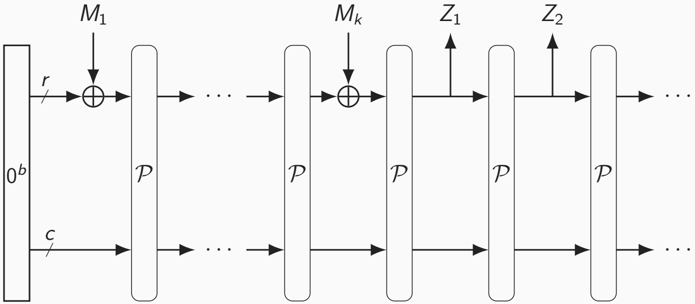
Radboud University (The Netherlands)

FSE

28 March 2024

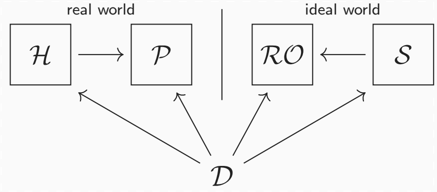
ESCADA

The Sponge Construction [Bertoni et al., 2007]



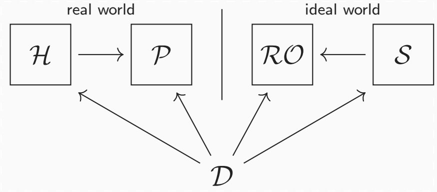
$M_1 \| \cdots \| M_k$ is the message padded into r-bit blocks

Indifferentiability [Maurer et al., 2004, Coron et al., 2005]



- $(\mathcal{H}^{\mathcal{P}}, \mathcal{P})$ for a random primitive $\mathcal{P}$ should behave like a random oracle $\mathcal{RO}$ paired with a simulator $\mathcal{S}$ that maintains construction-primitive consistency
- $\mathcal{H}$ is indifferentiable from $\mathcal{RO}$ for some simulator $\mathcal{S}$ whenever any $\mathcal{D}$ can distinguish the two worlds only with a negligible probability

Indifferentiability [Maurer et al., 2004, Coron et al., 2005]



- $(\mathcal{H}^{\mathcal{P}}, \mathcal{P})$ for a random primitive $\mathcal{P}$ should behave like a random oracle $\mathcal{RO}$ paired with a simulator $\mathcal{S}$ that maintains construction-primitive consistency
- $\mathcal{H}$ is indifferentiable from $\mathcal{RO}$ for some simulator $\mathcal{S}$ whenever any $\mathcal{D}$ can distinguish the two worlds only with a negligible probability
- Indifferentiability advantage:

$$\mathbf{Adv}^{\mathrm{iff}}_{\mathcal{H}}(q) = \max_{\mathcal{D} \text{ with } q \text{ queries}} \left| \mathbf{Pr}\left(\mathcal{D}^{\mathsf{Real}} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{Ideal}} = 1\right) \right|$$

Indifferentiability of the Sponge

- It has been proven that [Bertoni et al., 2008]

$$\mathbf{Adv}^{\mathrm{iff}}_{\mathsf{Sponge}}(q) = \mathcal{O}\left(\frac{q^2}{2^c}\right)$$

$\Longrightarrow$ The sponge is unlikely differentiable from a $\mathcal{RO}$ with less than $q \approx 2^{c/2}$ queries

- The bound is tight: finding collisions on the inner part allows to mount full-state collisions

Security of Keyed Sponges

- Keyed instances of the sponge may achieve security beyond $c/2$ bits
- Example: outer-keyed sponge

$$\mathrm{OKS}(K, M) = \mathrm{Sponge}(K \| M)$$

If K large enough, and online complexity $\sigma \ll 2^{c/2}$, OKS is secure up to $2^c/\sigma$ queries [Andreeva et al., 2015], [Naito and Yasuda, 2016], [Mennink, 2018]

- Keyed instances of the sponge may achieve security beyond $c/2$ bits
- Example: outer-keyed sponge

$$\mathrm{OKS}(K, M) = \mathrm{Sponge}(K \| M)$$

  If K large enough, and online complexity $\sigma \ll 2^{c/2}$, OKS is secure up to $2^c/\sigma$ queries [Andreeva et al., 2015], [Naito and Yasuda, 2016], [Mennink, 2018]
- One can go even further to 2^c security with Ascon-PRF [Dobraunig et al., 2021] (see [Mennink, 2023] for the exact statement) $\implies$ doubled security strength!

Motivation

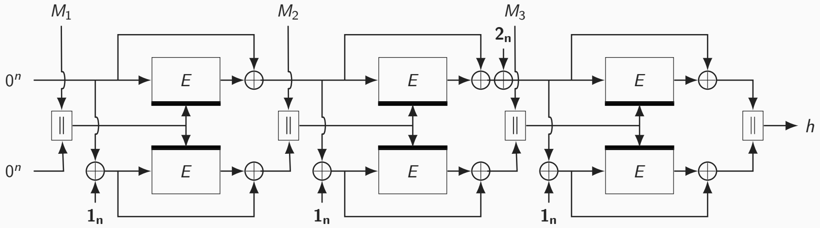
- Consider a permutation with size $b = 320$ (Ascon):
  - Sponge: up to 160 bits of security
  - Outer-keyed sponge: up to 270 bits of security (provided $\sigma < 2^{50}$)
- Smaller permutation sizes: consider Elephant [Beyne et al., 2020] NIST LWC finalist, based on permutations of sizes 160, 176, and 200 bits:
  - AEAD with at least 112 bits of security (provided $\sigma < 2^{50}$)
  - Sponge allows at most 100 bits of security $\implies$ no hashing functionality

- Consider a permutation with size $b = 320$ (Ascon):
  - Sponge: up to 160 bits of security
  - Outer-keyed sponge: up to 270 bits of security (provided $\sigma < 2^{50}$)
- Smaller permutation sizes: consider Elephant [Beyne et al., 2020] NIST LWC finalist, based on permutations of sizes 160, 176, and 200 bits:
  - AEAD with at least 112 bits of security (provided $\sigma < 2^{50}$)
  - Sponge allows at most 100 bits of security $\Longrightarrow$ no hashing functionality

Objective of this work: develop a permutation-based hashing construction with security beyond $b/2$ bits

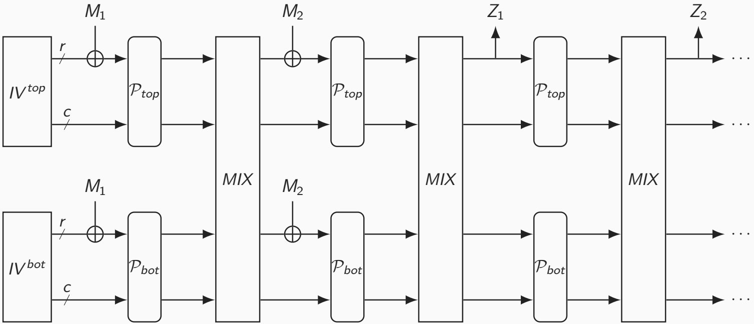
Double Block Length Hashing (DBLH)

- High-level idea: double the state size, call the primitive multiple times per compression function call
- Example: MDPH [Naito, 2019]

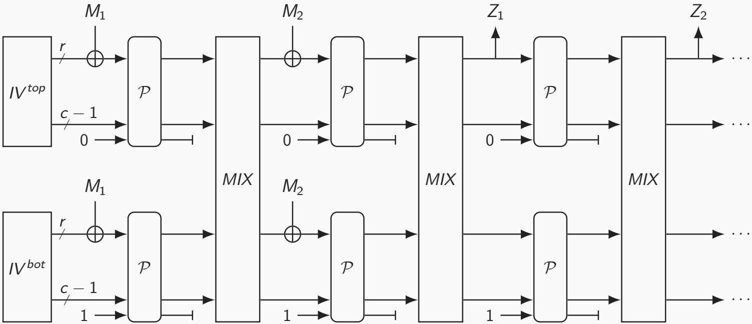


- Proven indifferentiable DBLH constructions are block cipher-based
- Block ciphers are *compressing* primitives, permutations are not

The Double Sponge



- The mixing layer is a simple MDS matrix: $MIX = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \in \mathcal{M}_{2\times 2}\left(GF(2^b)\right)$
- r bits absorbed/squeezed per compression function call



- The mixing layer is a simple MDS matrix: $MIX = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \in \mathcal{M}_{2\times 2}\left(GF(2^b)\right)$
- r bits absorbed/squeezed per compression function call
- Can use the same permutation at the top and bottom parts using domain separator bits

- We prove $2c/3$ bits of security:

$$\mathbf{Adv}^{\text{iff}}_{\mathcal{H}^P}(q) \le \frac{40q^{\frac{3}{2}}}{2^c - 3q}$$

$\implies$ Beyond the birthday bound in b when $3r \le c$

Security of the Double Sponge

- We prove $2c/3$ bits of security:

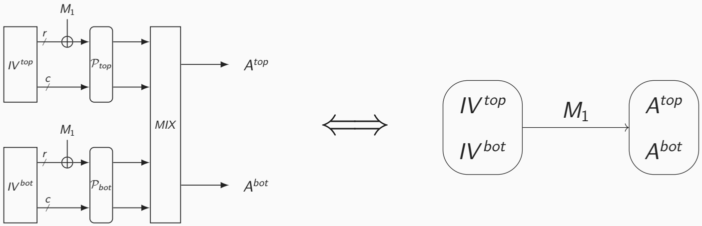
$$\mathbf{Adv}^{\mathrm{iff}}_{\mathcal{H}^P}(q) \leq \frac{40q^{\frac{3}{2}}}{2^c - 3q}$$

$\Longrightarrow$ Beyond the birthday bound in b when $3r \leq c$

$\Longrightarrow$ Can use smaller permutations, for a fixed level of security. For example:

  - $b = 176$ $(\mathrm{Spongent}-\pi[176])$ yields 112 bits of security with $r = 7$
  - $b = 200$ $(\mathrm{Keccak}-f[200])$ yields 112 bits of security with $r = 31$

Our Simulator

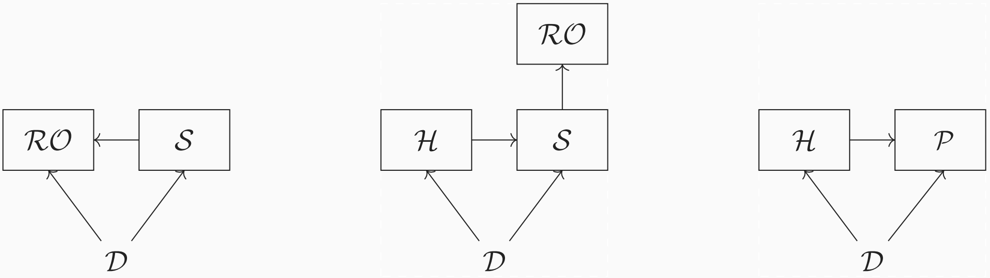


- Simulator $\mathcal{S}$ keeps track of the graph construction from its query history and ensures $\mathcal{RO}$ consistency as long as no bad event occurs
- $\mathcal{S}$ ensures that there exist no partial edge (i.e., $\mathcal{S}$ decides the image of $A^{top} \oplus (M_2 \| 0^c)$, but not of $A^{bot} \oplus (M_2 \| 0^c)$)

Similarly to [Naito and Ohta, 2014], an intermediate world is introduced:

$$\begin{tikzcd}
\boxed{\mathcal{RO}} & & \boxed{\mathcal{S}} \arrow[ll] \\
& \mathcal{D} \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & \boxed{\mathcal{RO}} \\
\boxed{\mathcal{H}} \arrow[rr] & & \boxed{\mathcal{S}} \arrow[u] \\
& \mathcal{D} \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\boxed{\mathcal{H}} \arrow[rr] & & \boxed{\mathcal{P}} \\
& \mathcal{D} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Similarly to [Naito and Ohta, 2014], an intermediate world is introduced:



$\mathcal{RO}$ consistency as long as no **BAD** $\implies$ indistinguishable

Statistical distance between $\mathcal{S}$ and $\mathcal{P}$ as long as no **BAD** $\implies \mathcal{O}\left(\frac{q^{3/2}}{2^c}\right)$

- Probability of **BAD**: $\mathcal{O}\left(\frac{q^{3/2}}{2^c}\right)$

Tightness of the Bound

- With respect to our simulator: attack in $2^{\frac{2c+r}{3}}$

$\Longrightarrow$ a gap of $r/3$ bits, likely lossy on the proof side

- With respect to any simulator: open question, designing a simulator that defeats the aforementioned attack and proving indifferentiability seems very hard
- We did not find a collision attack better than a “naive” one with cost $2^{c+r/2}$

Conclusion

- Double block length XOF construction:
  - Based on one b-bit permutation
  - Secure beyond $b/2$ bits given certain parameter sizes

$\Longrightarrow$ Allows to use smaller permutations for hashing

- Future work:
  - Close the gaps between security bound and attacks?
  - Explore alternative constructions?

Conclusion

- Double block length XOF construction:
  - Based on one b-bit permutation
  - Secure beyond $b/2$ bits given certain parameter sizes

$\Longrightarrow$ Allows to use smaller permutations for hashing

- Future work:
  - Close the gaps between security bound and attacks?
  - Explore alternative constructions?

Thank you for your attention!

Andreeva, E., Daemen, J., Mennink, B., and Assche, G. V. (2015).
Security of keyed sponge constructions using a modular proof approach.
In Leander, G., editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 364–384. Springer.

Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2007).
Sponge functions.
Ecrypt Hash Workshop 2007.

Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2008).
On the Indifferentiability of the Sponge Construction.
In Smart, N. P., editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer.

Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. (2020).
Dumbo, Jumbo, and Delirium: Parallel Authenticated Encryption for the Lightweight Circus.
IACR Trans. Symmetric Cryptol., 2020(S1):5–30.

Coron, J., Dodis, Y., Malinaud, C., and Puniya, P. (2005).
Merkle-Damgård Revisited: How to Construct a Hash Function.
In Shoup, V., editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer.

Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2021).
Ascon prf, mac, and short-input MAC.
IACR Cryptol. ePrint Arch., page 1574.

- Maurer, U. M., Renner, R., and Holenstein, C. (2004).
  Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology.
  In Naor, M., editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer.
- Mennink, B. (2018).
  Key Prediction Security of Keyed Sponges.
  IACR Trans. Symmetric Cryptol., 2018(4):128–149.
- Mennink, B. (2023).
  Understanding the Duplex and Its Security.
  IACR Trans. Symmetric Cryptol., 2023(2):1–46.

Naito, Y. (2019).
Optimally Indifferentiable Double-Block-Length Hashing Without Post-processing and with Support for Longer Key Than Single Block.
In Schwabe, P. and Thériault, N., editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, volume 11774 of *Lecture Notes in Computer Science*, pages 65–85. Springer.

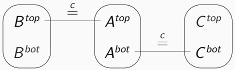
Naito, Y. and Ohta, K. (2014).
Improved Indifferentiable Security Analysis of PHOTON.
In Abdalla, M. and Prisco, R. D., editors, *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, volume 8642 of *Lecture Notes in Computer Science*, pages 340–357. Springer.

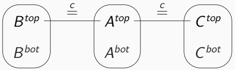
Naito, Y. and Yasuda, K. (2016).
New bounds for keyed sponges with extendable output: Independence between capacity and message length.
In Peyrin, T., editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 3–22. Springer.

Backup Slide: Bad Events

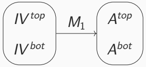
Three classes of bad events:

- Collision-taming:



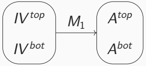


- Upper bounding $\mathcal{S}$'s query complexity and graph size:



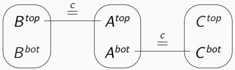
Top query with input $A^{top} \oplus (M_2 \| 0^c)$

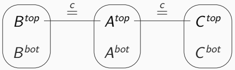
- $\mathcal{RO}$ consistency:



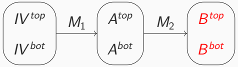
Three classes of bad events:

- Collision-taming:





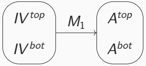
- Upper bounding $\mathcal{S}$'s query complexity and graph size:



Top query with input $A^{top} \oplus (M_2 \| 0^c)$

$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

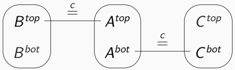
- $\mathcal{RO}$ consistency:

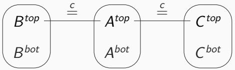


Backup Slide: Bad Events

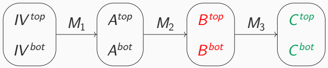
Three classes of bad events:

- Collision-taming:





- Upper bounding $\mathcal{S}$'s query complexity and graph size:

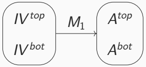


Top query with input $A^{top} \oplus (M_2 \| 0^c)$

$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

$C^{bot} \oplus (M_4 \| 0^c)$ already in $\mathcal{S}^{bot}$ query history

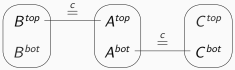
- $\mathcal{RO}$ consistency:

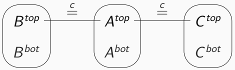


Backup Slide: Bad Events

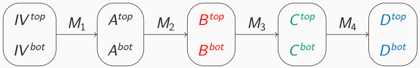
Three classes of bad events:

- Collision-taming:





- Upper bounding $\mathcal{S}$'s query complexity and graph size:

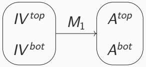


Top query with input $A^{top} \oplus (M_2 \| 0^c)$

$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

$C^{bot} \oplus (M_4 \| 0^c)$ already in $\mathcal{S}^{bot}$ query history

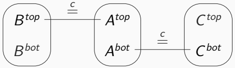
- $\mathcal{RO}$ consistency:

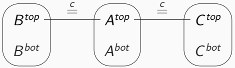


Backup Slide: Bad Events

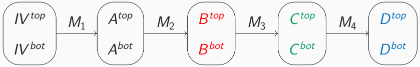
Three classes of bad events:

- Collision-taming:





- Upper bounding $\mathcal{S}$'s query complexity and graph size:

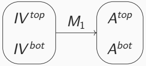


Top query with input $A^{top} \oplus (M_2 \| 0^c)$

$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

$C^{bot} \oplus (M_4 \| 0^c)$ already in $\mathcal{S}^{bot}$ query history

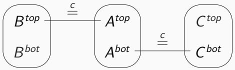
- $\mathcal{RO}$ consistency:

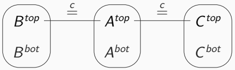


Backup Slide: Bad Events

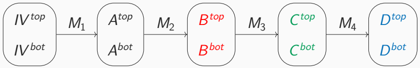
Three classes of bad events:

- Collision-taming:





- Upper bounding $\mathcal{S}$'s query complexity and graph size:

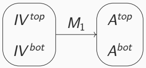


Top query with input $A^{top} \oplus (M_2\|0^c)$

$B^{top} \oplus (M_3\|0^c)$ already in $\mathcal{S}^{top}$ query history

$C^{bot} \oplus (M_4\|0^c)$ already in $\mathcal{S}^{bot}$ query history

- $\mathcal{RO}$ consistency:



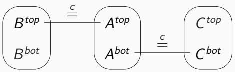
$B^{top} \oplus (M_3\|0^c)$ already in $\mathcal{S}^{top}$ query history

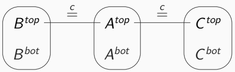
$B^{bot} \oplus (M_4\|0^c)$ already in $\mathcal{S}^{bot}$ query history

Backup Slide: Bad Events

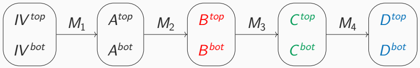
Three classes of bad events:

- Collision-taming:





- Upper bounding $\mathcal{S}$'s query complexity and graph size:

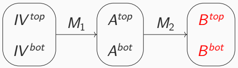


Top query with input $A^{top} \oplus (M_2 \| 0^c)$

$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

$C^{bot} \oplus (M_4 \| 0^c)$ already in $\mathcal{S}^{bot}$ query history

- $\mathcal{RO}$ consistency:



$B^{top} \oplus (M_3 \| 0^c)$ already in $\mathcal{S}^{top}$ query history

$B^{bot} \oplus (M_4 \| 0^c)$ already in $\mathcal{S}^{bot}$ query history