Squeezing the Most out of the Sponge

Charlotte Lefevre (based on joint works with Bart Mennink, Mario Marhuenda Beltrán, Siwei Sun, Shun Li, Zhiyu Zhang, Zhen Qin, Dengguo Feng)

Radboud University

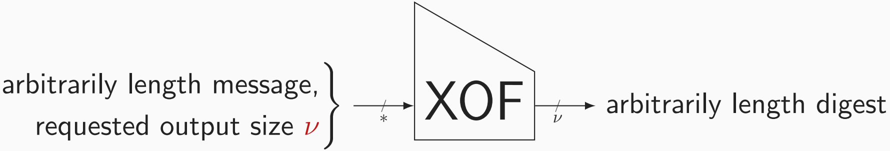
Nancy Seminar 2025

October 14, 2025

Overview i

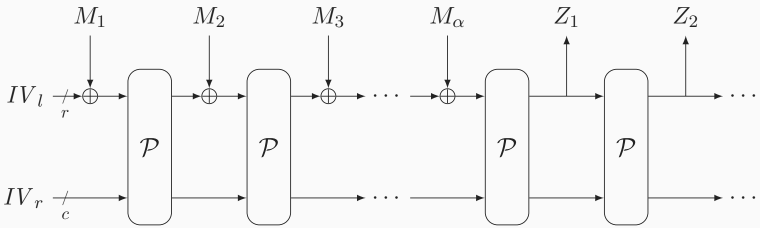
The Sponge Construction

Modern Definition of Hashing



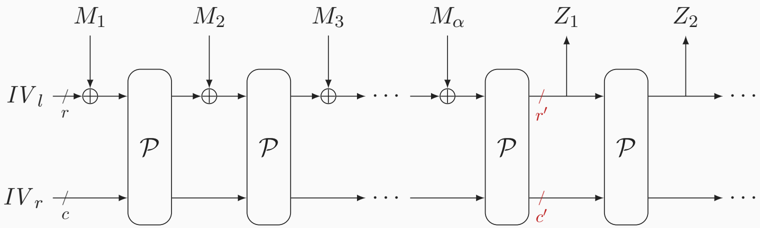
- Function XOF from $\{0,1\}^*$ to $\{0,1\}^\infty$
  - Variable-length input
  - Variable-length output
  - User specifies output length ν when calling the function

The Sponge Construction [BDPV07]



- State of size $b = r + c$ bits:
  - rate r (efficiency parameter)
  - capacity c (security parameter)
- $M_1 \| \cdots \| M_\alpha$ is the message padded into r-bit blocks (e.g., 10^* padding)
- SHA-3, Ascon-Hash

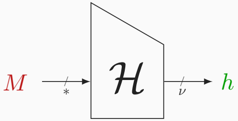
The Sponge Construction [BDPV07]



- State of size $b = r + c$ bits:
  - rate r (efficiency parameter)
  - capacity c (security parameter)
- $M_1 \| \cdots \| M_\alpha$ is the message padded into r-bit blocks (e.g., 10^* padding)
- SHA-3, Ascon-Hash
- Can squeeze at a larger rate [GPP11]

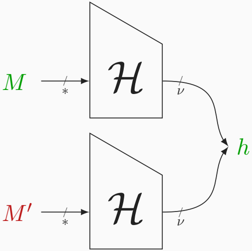
Classical security requirements

Preimage



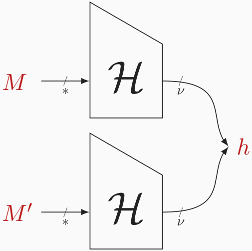
Given h, find M

Second Preimage



Given M, find $M' \neq M$

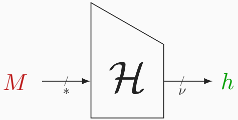
Collision



Find $M \neq M'$

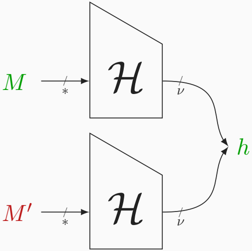
Classical security requirements

Preimage



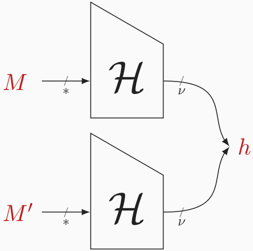
Given h, find M

Second Preimage



Given M, find $M' \neq M$

Collision

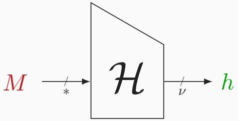


Find $M \neq M'$

- Consider the everywhere variants of (second) preimage [RS04] with $\mathcal{H}$ based on a random primitive.

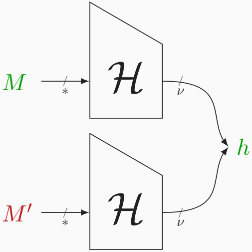
Classical security requirements

Preimage



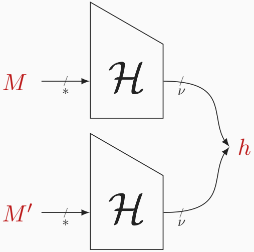
Given h, find M

Second Preimage



Given M, find $M' \neq M$

Collision

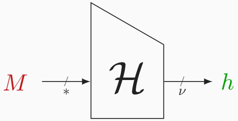


Find $M \neq M'$

- Consider the everywhere variants of (second) preimage [RS04] with $\mathcal{H}$ based on a random primitive.
- Not always sufficient, e.g., $\text{MAC}(k, m) = \mathcal{H}(k \| m)$ with $\mathcal{H} =$ plain MD

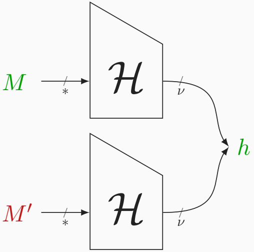
Classical security requirements

Preimage



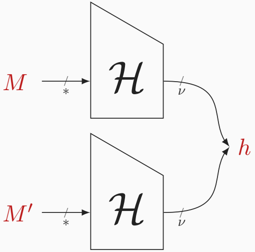
Given h, find M

Second Preimage



Given M, find $M' \neq M$

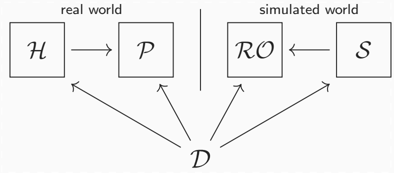
Collision



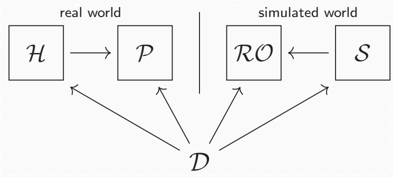
Find $M \neq M'$

- Consider the everywhere variants of (second) preimage [RS04] with $\mathcal{H}$ based on a random primitive.
- Not always sufficient, e.g., $\mathrm{MAC}(k, m) = \mathcal{H}(k \| m)$ with $\mathcal{H} =$ plain MD
- Hash function should behave like a random oracle

Indifferentiability [MRH04, CDMP05]

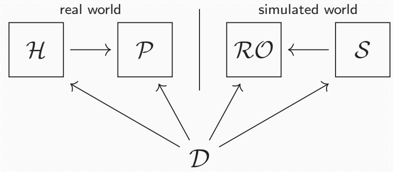


- $(\mathcal{H}^{\mathcal{P}}, \mathcal{P})$ for a random primitive $\mathcal{P}$ should behave like a random oracle $\mathcal{RO}$ paired with a simulator $\mathcal{S}$ that maintains construction-primitive consistency
- $\mathcal{H}$ is indifferentiable from $\mathcal{RO}$ for some simulator $\mathcal{S}$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



- $(\mathcal{H}^{\mathcal{P}}, \mathcal{P})$ 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
- In our case: $\mathcal{P}$ is a random permutation; let $\mathcal{N}$ be the number of $\mathcal{P}$-calls that the queries would induce in the real world

Indifferentiability [MRH04, CDMP05]



- $(\mathcal{H}^{\mathcal{P}}, \mathcal{P})$ 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
- In our case: $\mathcal{P}$ is a random permutation; let $\mathcal{N}$ be the number of $\mathcal{P}$-calls that the queries would induce in the real world

$\Longrightarrow$ For $\mathsf{atk} \in \{\mathsf{col}, \mathsf{sec}, \mathsf{pre}\}$, we have [AMP10]

$$\mathbf{Adv}_{\mathcal{H}}^{\mathrm{atk}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{H}}^{\mathrm{iff}}(\mathcal{A}') + \mathbf{Adv}_{\mathcal{RO}}^{\mathrm{atk}}(\mathcal{A}'')$$

Indifferentiability of the Sponge Construction

- The sponge construction was proven indifferentiable with a tight bound [BDPV08]

$$\mathbf{Adv}^{\mathrm{iff}}_{\mathrm{Sponge}}(\mathcal{N}) \leq \frac{\mathcal{N}(\mathcal{N}+1)}{2^c}$$

Indifferentiability of the Sponge Construction

- The sponge construction was proven indifferentiable with a tight bound [BDPV08]

$$\mathbf{Adv}^{\mathrm{iff}}_{\mathrm{Sponge}}(\mathcal{N}) \leq \frac{\mathcal{N}(\mathcal{N}+1)}{2^c}$$

- The generalized sponge construction was proven indifferentiable with bound [NO14]

$$\tilde{\mathcal{O}}\left(\frac{\mathcal{N}}{2^{c'}} + \frac{\mathcal{N}(\mathcal{N}+1)}{2^c}\right)$$

$\implies$ c' may be set to $\approx c/2$ bits

Collision and (Second) Preimage Resistance of the Sponge

- Security of sponge truncated to ν bits against classical attacks [AMP10] (assuming $c' \geq c/2$):

Collision resistance:	$\mathcal{N}^2/2^c + \mathcal{N}^2/2^{\nu+1}$
Second preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^{\nu}$
Preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^{\nu}$

Collision and (Second) Preimage Resistance of the Sponge

- Security of sponge truncated to ν bits against classical attacks [AMP10] (assuming $c' \geq c/2$):

Collision resistance:	$\mathcal{N}^2/2^c + \mathcal{N}^2/2^{\nu+1}$
Second preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^\nu$
Preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^\nu$

$\uparrow$ distance from sponge to RO ($\mathcal{N}$ is # primitive evaluations)

$\uparrow$ classical attacks against RO ($\mathcal{N}$ is # oracle evaluations)

Collision and (Second) Preimage Resistance of the Sponge

- Security of sponge truncated to ν bits against classical attacks [AMP10] (assuming $c' \geq c/2$):

Collision resistance:	$\mathcal{N}^2/2^c + \mathcal{N}^2/2^{\nu+1}$	$\leftarrow$ attack in $\min\{2^{c/2}, 2^{\nu/2}\}$
Second preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^{\nu}$	$\leftarrow$ attack in $\min\{2^{c/2}, 2^{\nu}\}$
Preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^{\nu}$	$\leftarrow$ attack in $\min\{2^{\nu-r'} + 2^{c/2}, 2^{\nu}\}$

$\uparrow$ distance from sponge to RO ($\mathcal{N}$ is $\#$ primitive evaluations)

$\uparrow$ classical attacks against RO ($\mathcal{N}$ is $\#$ oracle evaluations)

- Attacks already described in [BDPV07]

Collision and (Second) Preimage Resistance of the Sponge

- Security of sponge truncated to ν bits against classical attacks [AMP10] (assuming $c' \geq c/2$):

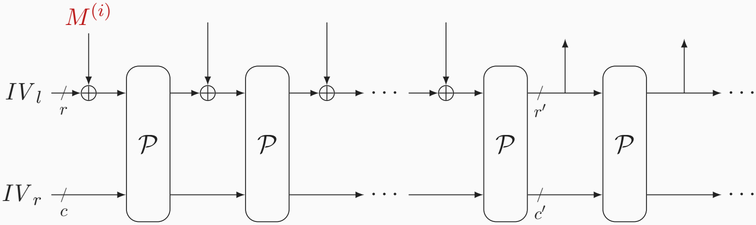
Collision resistance:	$\mathcal{N}^2/2^c + \mathcal{N}^2/2^{\nu+1}$	$\leftarrow$ attack in $\min\{2^{c/2}, 2^{\nu/2}\}$
Second preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^\nu$	$\leftarrow$ attack in $\min\{2^{c/2}, 2^{\nu}\}$
Preimage resistance:	$\mathcal{N}^2/2^c + \mathcal{N}/2^\nu$	$\leftarrow$ attack in $\min\{2^{\nu-r'} + 2^{c/2}, 2^{\nu}\}$

distance from sponge to RO ($\mathcal{N}$ is # primitive evaluations)

classical attacks against RO ($\mathcal{N}$ is # oracle evaluations)

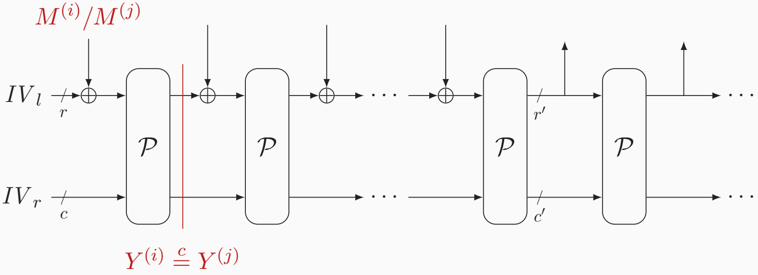
- Attacks already described in [BDPV07]
- In [LM22], this gap was closed

Collision Attack on the Sponge [BDPV11a]

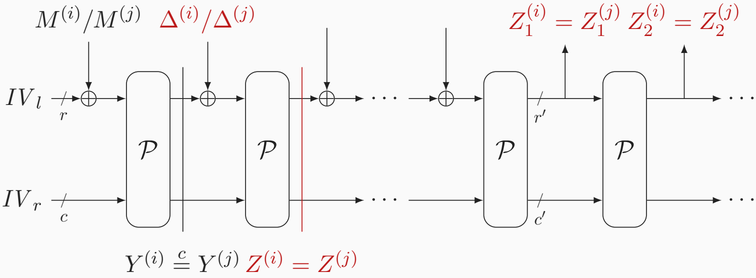


- Absorb $2^{c/2}$ different messages

Collision Attack on the Sponge [BDPV11a]

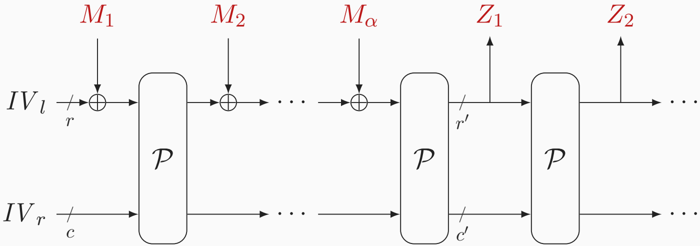


- Absorb $2^{c/2}$ different messages
- With high probability, there exists $Y^{(i)} \neq Y^{(j)}$ that collide on their inner part

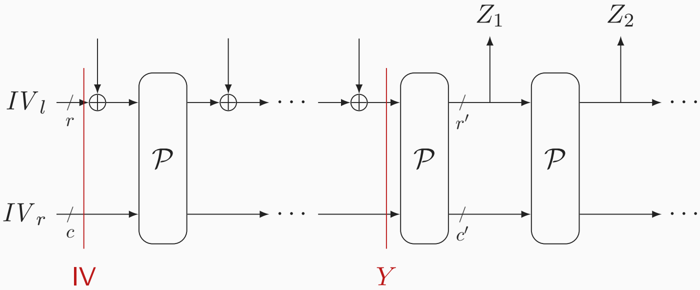


- Absorb $2^{c/2}$ different messages
- With high probability, there exists $Y^{(i)} \neq Y^{(j)}$ that collide on their inner part
- Compensate the difference in next absorb call
- Triggers a full state collision $\implies$ collision in digests

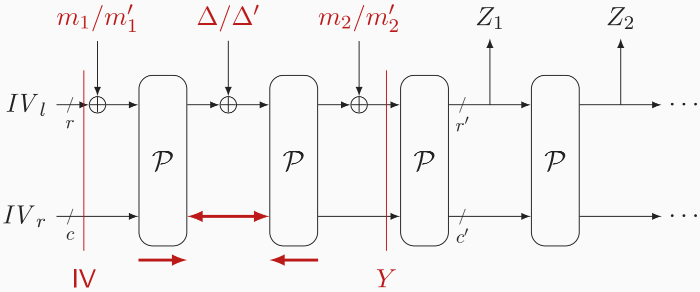
Second Preimage Attack on the Sponge [BDPV11a]



- Let $M_1 \| M_2 \| \cdots \| M_\alpha$ be the first preimage blocks

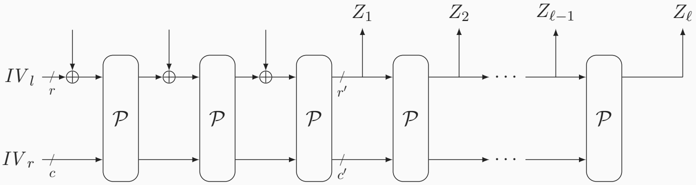


- Let $M_1 \| M_2 \| \cdots \| M_\alpha$ be the first preimage blocks
- Retrieve Y, the state right before squeezing



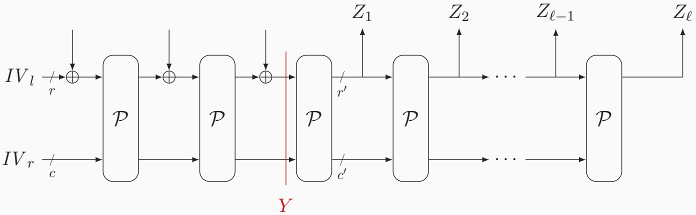
- Let $M_1 \| M_2 \| \cdots \| M_\alpha$ be the first preimage blocks
- Retrieve Y, the state right before squeezing
- Connect the Y and IV with an inner collision
- Cost: $\approx 2^{c/2}$

First Preimage Attack on the Sponge [BDPV11a]



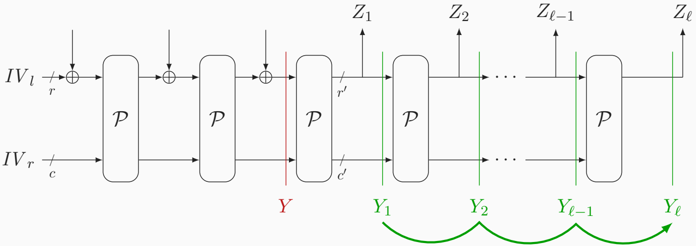
- Let $Z_1 \| Z_2 \| \cdots \| Z_\ell$ be the image

First Preimage Attack on the Sponge [BDPV11a]



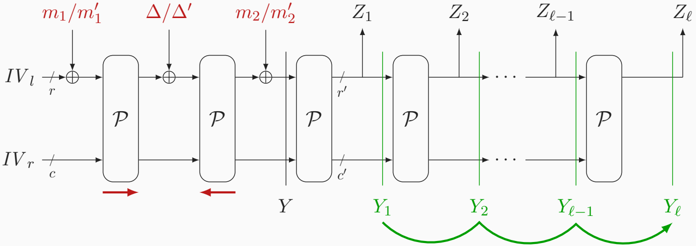
- Let $Z_1 \| Z_2 \| \cdots \| Z_\ell$ be the image
- No intermediate state Y is given: need to find it

First Preimage Attack on the Sponge [BDPV11a]



- Let $Z_1 \| Z_2 \| \cdots \| Z_\ell$ be the image
- No intermediate state Y is given: need to find it
- Start from a Y_1 with correct outerpart, make $\ell - 1$ forward queries: one attempt succeeds with probability $\approx \frac{1}{2^{\nu - r'}}$

First Preimage Attack on the Sponge [BDPV11a]



- Let $Z_1 \| Z_2 \| \cdots \| Z_\ell$ be the image
- No intermediate state Y is given: need to find it
- Start from a Y_1 with correct outerpart, make $\ell - 1$ forward queries: one attempt succeeds with probability $\approx \frac{1}{2^{\nu - r'}}$
- Total attack costs $\approx 2^{\nu - r'} + 2^{c/2}$ queries

Improved First Preimage Resistance

- Best known attack has cost $\approx \min\left\{2^{\nu}, 2^{\nu-r'} + 2^{c/2}\right\}$
- Indifferentiability guarantees security up to $\approx \min\left\{2^{\nu}, 2^{c/2}\right\}$ queries

Improved First Preimage Resistance

- Best known attack has cost $\approx \min\left\{2^{\nu}, 2^{\nu-r'} + 2^{c/2}\right\}$
- Indifferentiability guarantees security up to $\approx \min\left\{2^{\nu}, 2^{c/2}\right\}$ queries

$\Longrightarrow$ Bound is not tight when $c/2 \leq \nu - r'$

Improved First Preimage Resistance

- Best known attack has cost $\approx \min\left\{2^{\nu}, 2^{\nu-r'} + 2^{c/2}\right\}$
- Indifferentiability guarantees security up to $\approx \min\left\{2^{\nu}, 2^{c/2}\right\}$ queries

$\implies$ Bound is not tight when $c/2 \leq \nu - r'$

- [LM22]: preimage resistance proven with bound

$$\tilde{\mathcal{O}}\left(\frac{\mathcal{N}}{2^{\nu}} + \min\left\{\frac{\mathcal{N}}{2^{\nu-r'}}, \frac{\mathcal{N}^2}{2^{c}}\right\}\right)$$

$\implies$ Optimality of the attack

Improved First Preimage Resistance

- Best known attack has cost $\approx \min\left\{2^{\nu}, 2^{\nu-r'} + 2^{c/2}\right\}$
- Indifferentiability guarantees security up to $\approx \min\left\{2^{\nu}, 2^{c/2}\right\}$ queries

$\Longrightarrow$ Bound is not tight when $c/2 \leq \nu - r'$

- [LM22]: preimage resistance proven with bound

$$\tilde{\mathcal{O}}\left(\frac{\mathcal{N}}{2^{\nu}} + \min\left\{\frac{\mathcal{N}}{2^{\nu-r'}}, \frac{\mathcal{N}^2}{2^{c}}\right\}\right)$$

$\Longrightarrow$ Optimality of the attack

- The proof relies on decomposing the bad events triggered by adversary by following the aforementioned attack

Application

- No impact on SHA-3 members, as they squeeze in one block
- Ascon-Hash parameters:

Application

- No impact on SHA-3 members, as they squeeze in one block
- Ascon-Hash parameters:
  - $(b, c, r, \nu) = (320, 256, 64, 256)$
  - Generic preimage resistance improved from 128 to 192 bits of security
- Other lightweight sponges may also benefit (e.g., NIST LWC finalists, Spongent, Photon)

Permutation-Based Hashing with Stronger (Second) Preimage Resistance

Stronger (Second) Preimage Resistance

- Some applications require strong security guarantees, such as hash-based post-quantum signature schemes

Stronger (Second) Preimage Resistance

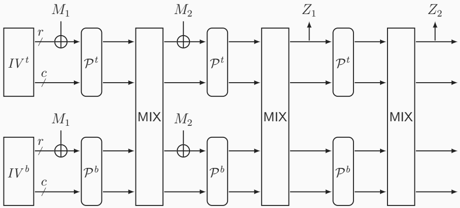
- Some applications require strong security guarantees, such as hash-based post-quantum signature schemes
- Example: the National Institute of Commercial Cryptography Standards (NICCS) of China calls for hash functions with 1024-bit preimage and second-preimage resistance

Stronger (Second) Preimage Resistance

- Some applications require strong security guarantees, such as hash-based post-quantum signature schemes
- Example: the National Institute of Commercial Cryptography Standards (NICCS) of China calls for hash functions with 1024-bit preimage and second-preimage resistance
  - To achieve this with a sponge, the capacity must be at least 2048 bits
  - As a result, Keccak-p[1600,24] is not suitable for this scenario

Double Sponge

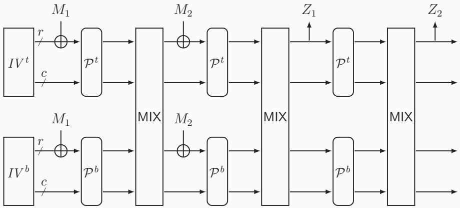
- One possibility: use the double sponge construction [LM24]



- Indifferentiable up to $\approx 2^{\frac{2c}{3}}$ queries

Double Sponge

- One possibility: use the double sponge construction [LM24]



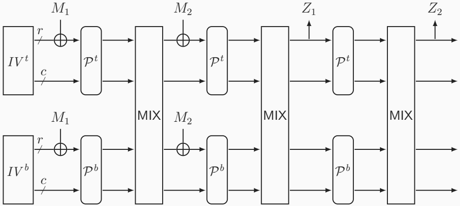
- Indifferentiable up to $\approx 2^{\frac{2c}{3}}$ queries

$\implies$ With Keccak-p[1600,24] ($c = 1536, r = 64$), we get 1024 bits of indifferentiability security

- Not the most efficient approach

Double Sponge

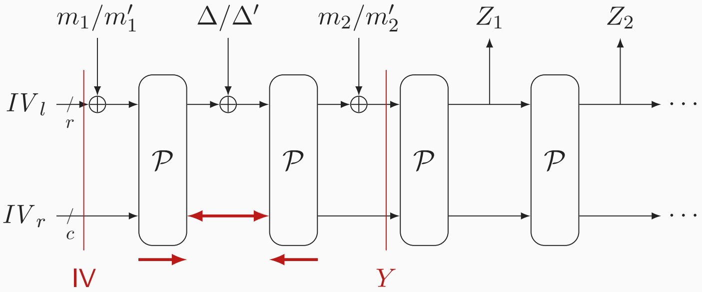
- One possibility: use the double sponge construction [LM24]



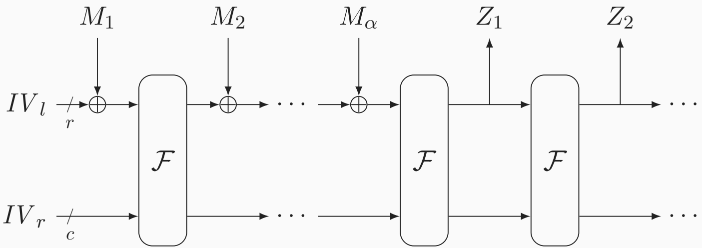
- Indifferentiable up to $\approx 2^{\frac{2c}{3}}$ queries

$\Longrightarrow$ With Keccak-p[1600,24] ($c = 1536, r = 64$), we get 1024 bits of indifferentiability security

- Not the most efficient approach
- Instead, aim to achieve λ-bit security for both preimage and second preimage using a sponge with λ-bit capacity

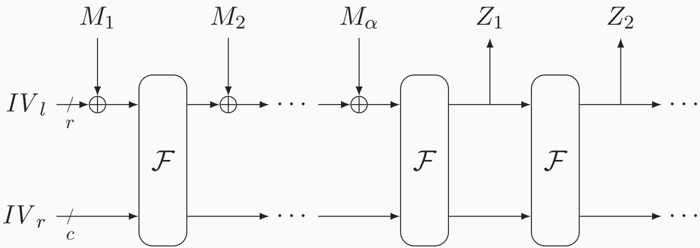


- The attacks on sponge for (second) preimage rely on the invertibility of $\mathcal{P}$



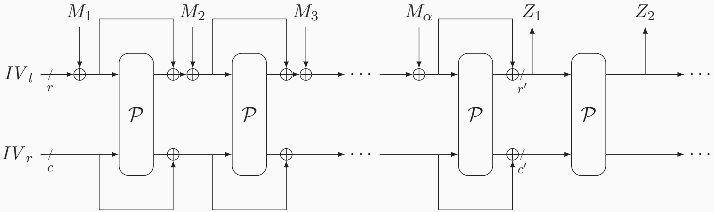
- The attacks on sponge for (second) preimage rely on the invertibility of $\mathcal{P}$
- Foekens [Foe23] proved that sponge with a random transformation has:
  - Preimage resistance up to $\approx 2^{\nu}$ queries
  - Second preimage resistance up to $\approx \min\{2^{\nu}, 2^{c}/\alpha\}$ queries (α: size of first preimage)

Sponge with a Random Transformation



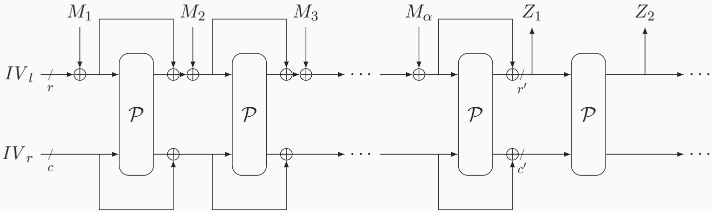
- The attacks on sponge for (second) preimage rely on the invertibility of $\mathcal{P}$
- Foekens [Foe23] proved that sponge with a random transformation has:
  - Preimage resistance up to $\approx 2^{\nu}$ queries
  - Second preimage resistance up to $\approx \min\{2^{\nu}, 2^{c}/\alpha\}$ queries (α: size of first preimage)
- But non-invertible functions more scarce than permutations

Sponge with a Feed-Forward: SPONGE-DM



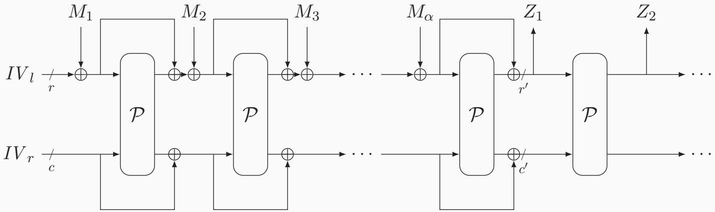
- **Alternative**: add a feed-forward only during absorption (independent related work: [GHF$^+$25])

Sponge with a Feed-Forward: SPONGE-DM



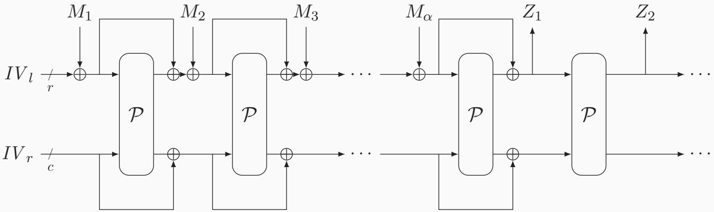
- **Alternative**: add a feed-forward only during absorption (independent related work: [GHF⁺25])
- This construction, named SPONGE-DM, achieves
  - Preimage resistance up to $\approx 2^\nu$ queries
  - Second preimage resistance up to $\min\{2^\nu, 2^c/\alpha\}$ queries
  - Indifferentiability up to sponge's bound

Sponge with a Feed-Forward: SPONGE-DM



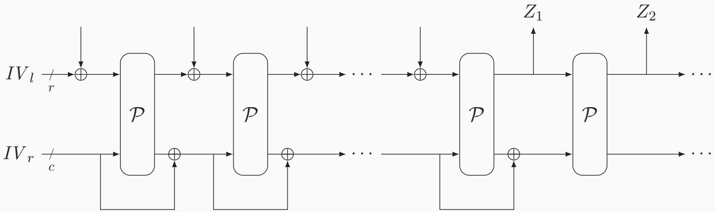
- **Alternative**: add a feed-forward only during absorption (independent related work: [GHF+25])
- This construction, named SPONGE-DM, achieves
  - Preimage resistance up to $\approx 2^\nu$ queries
  - Second preimage resistance up to $\min\{2^\nu, 2^c/\alpha\}$ queries
  - Indifferentiability up to sponge's bound
- **Cost**: adds a extra b-bit state $\implies$ can we lower the size of the feed-forward?

Sponge with a Feed-Forward: Attempt to Reduce Size of Feed-Forward



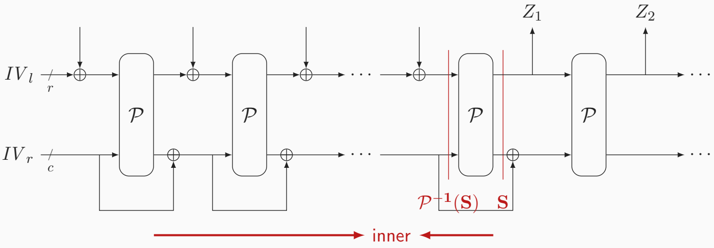
- Intuition: the adversary has control on the outer part of the state

Sponge with a Feed-Forward: Attempt to Reduce Size of Feed-Forward



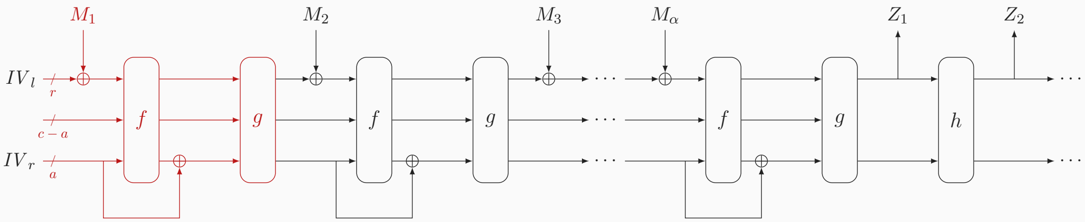
- Intuition: the adversary has control on the outer part of the state, so feeding it forward seems unnecessary

Sponge with a Feed-Forward: Attempt to Reduce Size of Feed-Forward



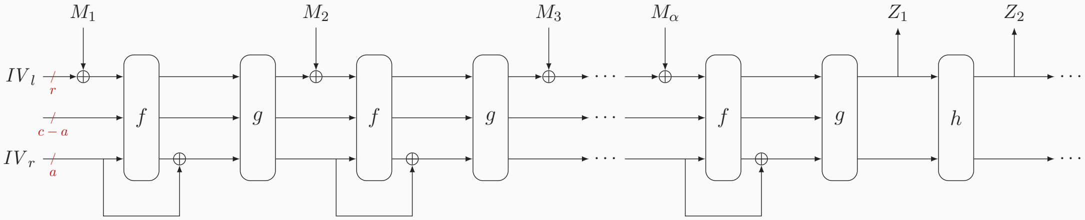
- Intuition: the adversary has control on the outer part of the state, so feeding it forward seems unnecessary
- This intuition is incorrect: can mount to attack in $\approx 2^{c/2}$ queries when $n \leq r$

SPONGE-EDM^a for $a = \{0, \ldots, b\}$



- Idea: augment absorbing phase by one extra permutation call (may have $a \geq c$)
- May use round-reduced permutations

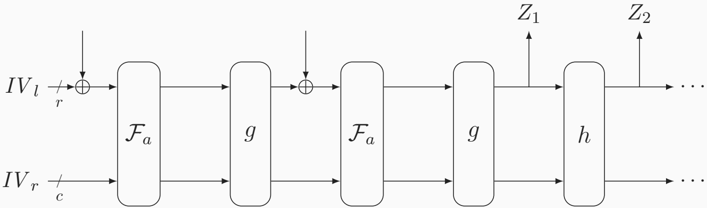
SPONGE-EDM^a for $a = \{0, \ldots, b\}$



- Idea: augment absorbing phase by one extra permutation call (may have $a \geq c$)
- May use round-reduced permutations
- We proved:
  - $\approx \min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$ bits of preimage security
  - $\approx \min\{\nu, c - \log_2(\alpha), \max\{a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$ bits of second preimage security

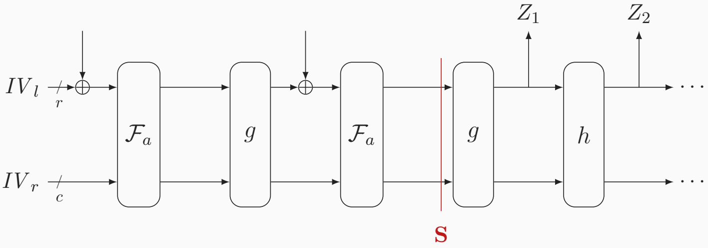
$\mathcal{F}_a(x) = f(x) \oplus 0^{b-a} \| \lfloor x \rfloor_a$

security: $\min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$



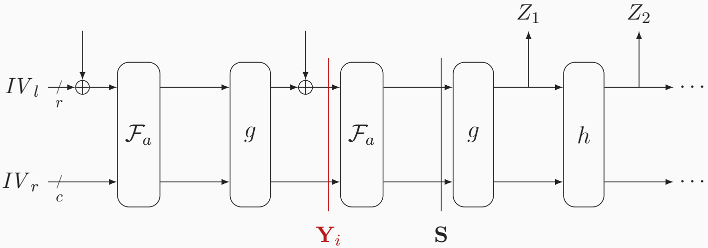
$\mathcal{F}_a(x) = f(x) \oplus 0^{b-a} \| \lfloor x \rfloor_a$

security: $\min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$



❶ Find a squeezing state $\mathbf{S}$ (cost: $2^{\nu - r'}$)

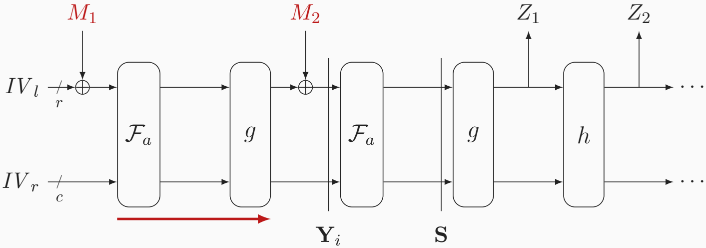
$\mathcal{F}_a(x) = f(x) \oplus 0^{b-a} \| \lfloor x \rfloor_a$ security: $\min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$



1. Find a squeezing state $\mathbf{S}$ (cost: $2^{\nu-r'}$)
2. Invert $\mathcal{F}_a$ several times (costs 2^a per inversion, do $2^{\min\{0,\frac{c-a}{2}\}}$ inversions)

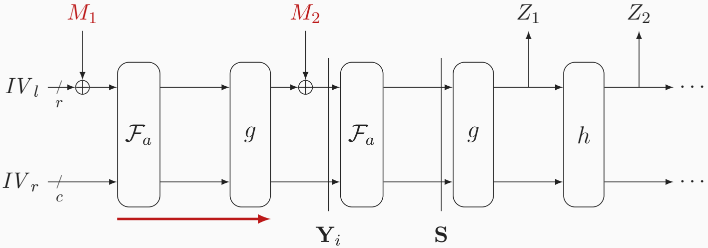
$\mathcal{F}_a(x) = f(x) \oplus 0^{b-a} \| \lfloor x \rfloor_a$

security: $\min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$



1. Find a squeezing state $\mathbf{S}$ (cost: $2^{\nu - r'}$)
2. Invert $\mathcal{F}_a$ several times (costs 2^a per inversion, do $2^{\min\{0, \frac{c-a}{2}\}}$ inversions)
3. Find an inner collision between some $\mathbf{Y}_i$ and a forward query (cost: $\min\{2^c, 2^{\frac{a+c}{2}}\}$)

$\mathcal{F}_a(x) = f(x) \oplus 0^{b-a} \| \lfloor x \rfloor_a$ security: $\min\{\nu, \max\{\nu - r', a, \min\{\frac{a+c}{2}, \frac{b+c}{3}\}\}\}$



1. Find a squeezing state $\mathbf{S}$ (cost: $2^{\nu-r'}$)
2. Invert $\mathcal{F}_a$ several times (costs 2^a per inversion, do $2^{\min\{0,\frac{c-a}{2}\}}$ inversions)
3. Find an inner collision between some $\mathbf{Y}_i$ and a forward query (cost: $\min\{2^c, 2^{\frac{a+c}{2}}\}$)
4. Overall cost: $2^{\nu-r'} + \max\{2^a, 2^{\frac{a+c}{2}}\}$

Application

- Using Keccak-p[1600] inside $\texttt{SPONGE-EDM}^c$ with $c = 1088$, $\nu = 1024$ (f and g with each 12 rounds) meets NICCS requirements:

Application

- Using Keccak-p[1600] inside $\texttt{SPONGE-EDM}^c$ with $c = 1088$, $\nu = 1024$ (f and g with each 12 rounds) meets NICCS requirements:
  - generic collision resistance ≈ 512 bits
  - generic preimage resistance ≈ 1024 bits
  - generic second-preimage resistance ≈ 1024 bits assuming $\alpha < 2^{64}$

Application

- Using Keccak-p[1600] inside $\texttt{SPONGE-EDM}^c$ with $c = 1088$, $\nu = 1024$ (f and g with each 12 rounds) meets NICCS requirements:
  - generic collision resistance ≈ 512 bits
  - generic preimage resistance ≈ 1024 bits
  - generic second-preimage resistance ≈ 1024 bits assuming $\alpha < 2^{64}$
- Ascon-Sign (PQC submission) uses Ascon-Hash within SPHINCS+ framework: (Ascon-Hash parameters: $(b, c, r, \nu) = (320, 256, 64, 256)$)

Application

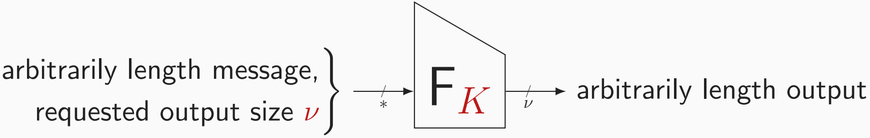
- Using Keccak-p[1600] inside $\texttt{SPONGE-EDM}^c$ with $c = 1088$, $\nu = 1024$ (f and g with each 12 rounds) meets NICCS requirements:
  - generic collision resistance ≈ 512 bits
  - generic preimage resistance ≈ 1024 bits
  - generic second-preimage resistance ≈ 1024 bits assuming $\alpha < 2^{64}$
- Ascon-Sign (PQC submission) uses Ascon-Hash within SPHINCS+ framework: (Ascon-Hash parameters: $(b, c, r, \nu) = (320, 256, 64, 256)$)
  - Submission claimed 192-bit security, but generic second preimage resistance at best $c/2 = 128$ bits $\implies$ did not advance second round
  - Instantiating the hash construction with $\texttt{SPONGE-DM}$ remedies this gap $\implies$ 192-bit generic security

Application

- Using Keccak-p[1600] inside $\texttt{SPONGE-EDM}^c$ with $c = 1088$, $\nu = 1024$ (f and g with each 12 rounds) meets NICCS requirements:
  - generic collision resistance ≈ 512 bits
  - generic preimage resistance ≈ 1024 bits
  - generic second-preimage resistance ≈ 1024 bits assuming $\alpha < 2^{64}$
- Ascon-Sign (PQC submission) uses Ascon-Hash within SPHINCS+ framework: (Ascon-Hash parameters: $(b, c, r, \nu) = (320, 256, 64, 256)$)
  - Submission claimed 192-bit security, but generic second preimage resistance at best $c/2 = 128$ bits $\implies$ did not advance second round
  - Instantiating the hash construction with $\texttt{SPONGE-DM}$ remedies this gap $\implies$ 192-bit generic security
- Future work: what about the setting of a quantum adversary?

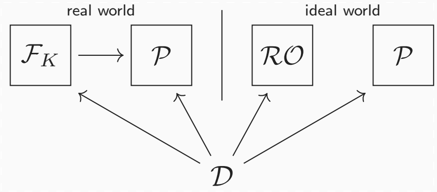
Sponge-Based PRFs

Pseudorandom Function (PRF)



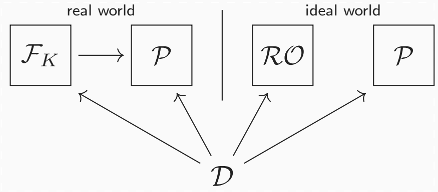
- Keyed function F_K from $\{0,1\}^*$ to $\{0,1\}^\infty$
  - Variable-length input
  - Ideally: variable-length output, where user specifies output length ν when calling the function

Indistinguishability in the Ideal Model



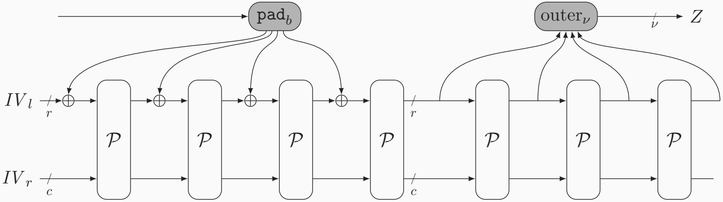
- $\mathcal{F}_K^{\mathcal{P}}$ for a random primitive $\mathcal{P}$ and key K should behave like a random oracle $\mathcal{RO}$

Indistinguishability in the Ideal Model



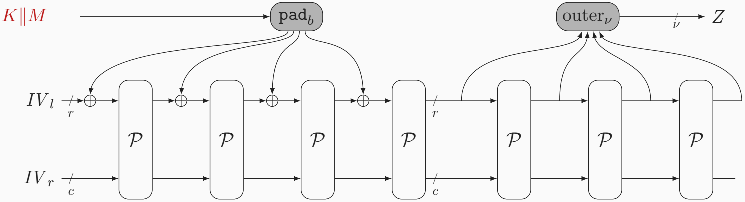
- $\mathcal{F}_K^{\mathcal{P}}$ for a random primitive $\mathcal{P}$ and key K should behave like a random oracle $\mathcal{RO}$
- In our case: $\mathcal{P}$ is a random permutation, and let:
  - $\mathcal{N}$ number of $\mathcal{P}$-queries,
  - $\mathcal{M}$ online complexity (number of blocks),
  - μ number of users

Turning a Sponge into a PRF



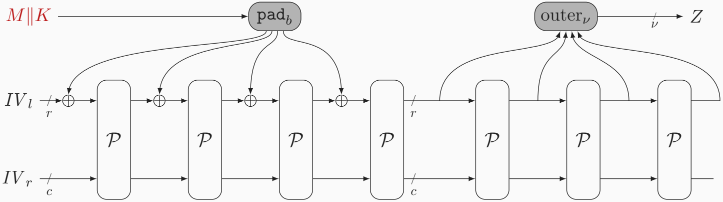
- Black-box approaches to keying a sponge:

Turning a Sponge into a PRF



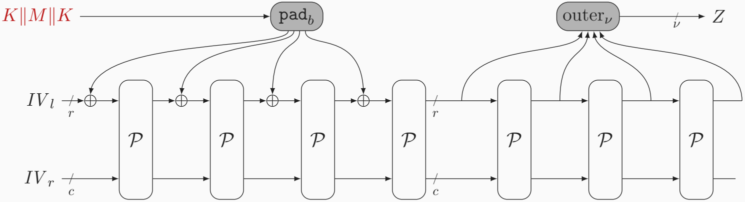
- Black-box approaches to keying a sponge:
- Outer-Keyed Sponge (OKS) [BDPV11b]

Turning a Sponge into a PRF

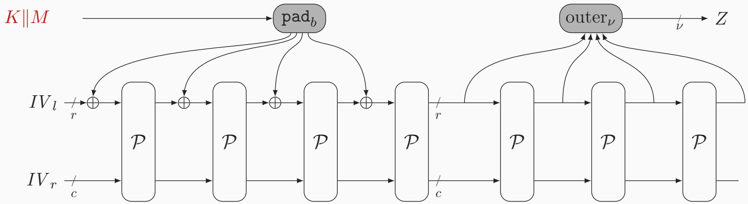


- Black-box approaches to keying a sponge:
- Outer-Keyed Sponge (OKS) [BDPV11b]
- Suffix-Keyed Sponge [BDPV11a, DM19, DM20]

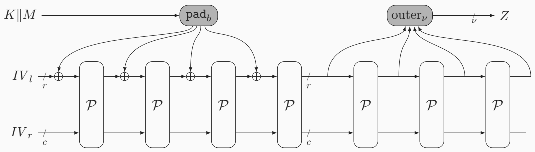
Turning a Sponge into a PRF



- Black-box approaches to keying a sponge:
- Outer-Keyed Sponge (OKS) [BDPV11b]
- Suffix-Keyed Sponge [BDPV11a, DM19, DM20]
- Sandwich Keyed Sponge [Nai16]



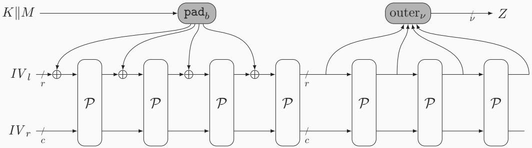
- Black-box approaches to keying a sponge:
- Outer-Keyed Sponge (OKS) [BDPV11b]
- Suffix-Keyed Sponge [BDPV11a, DM19, DM20]
- Sandwich Keyed Sponge [Nai16]
- These variants give different security guarantees; here we focus on OKS



- Indifferentiability is overkill, dedicated proofs give tighter security bounds: assuming that $\lceil k/r \rceil$ is a small constant, we have [GPT15, ADMV15, NY16, Men18]

$$\mathbf{Adv}_{\text{OKS}}^{\mu\text{-PRF}} = \tilde{\mathcal{O}}\left(\frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^c}\right)$$

OKS: Security

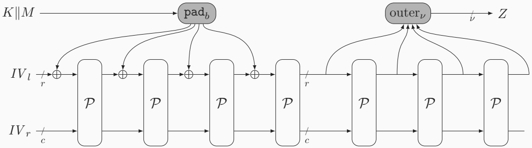


- Indifferentiability is overkill, dedicated proofs give tighter security bounds: assuming that $\lceil k/r \rceil$ is a small constant, we have [GPT15, ADMV15, NY16, Men18]

$$\mathbf{Adv}_{\mathsf{OKS}}^{\mu\text{-}\mathsf{PRF}} = \tilde{\mathcal{O}}\left(\frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^c}\right)$$

- The bound is tight

OKS: Security

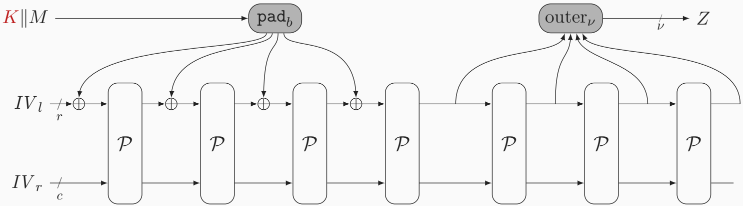


- Indifferentiability is overkill, dedicated proofs give tighter security bounds: assuming that $\lceil k/r \rceil$ is a small constant, we have [GPT15, ADMV15, NY16, Men18]

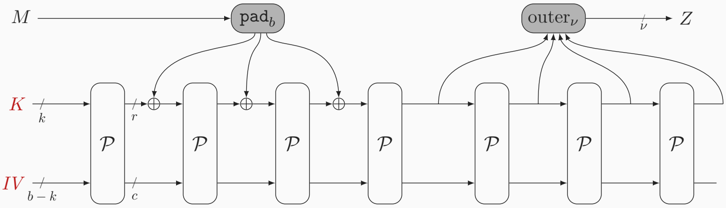
$$\mathbf{Adv}^{\mu\text{-PRF}}_{\mathsf{OKS}} = \tilde{\mathcal{O}}\left(\frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^c}\right)$$

- The bound is tight
- **Concrete example:** with $b = 320$, $r = 128$, $c = 192$, $k = 128$, $\mu = 1$, and assuming $\mathcal{M} \ll 2^{64}$, this gives 128 bits of security

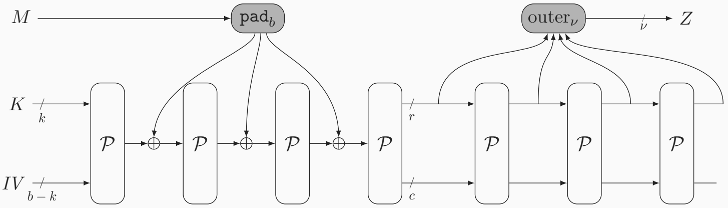
OKS: Improvements [BDPV11b, BDPV12, CDH+12, ADMV15]



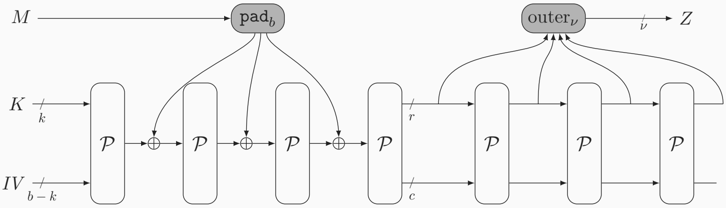
OKS: Improvements [BDPV11b, BDPV12, CDH+12, ADMV15]



- V2: Key into the initial state $\implies$ more efficient, no security loss



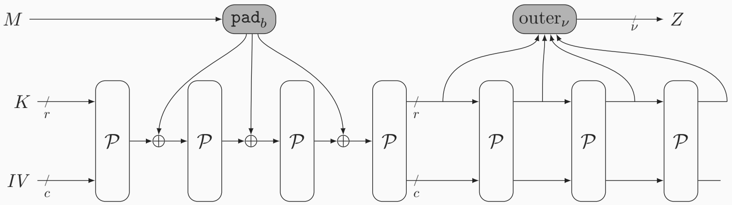
- V2: Key into the initial state $\implies$ more efficient, no security loss
- V3: absorb over the entire state $\implies$ Full-State Keyed Sponge (FSKS) [MRV15]



- V2: Key into the initial state $\implies$ more efficient, no security loss
- V3: absorb over the entire state $\implies$ Full-State Keyed Sponge (FSKS) [MRV15]
- With some optimizations (e.g., unique IV per user, domain separation), security can be pushed as far as [DM24, DMV17, Men23]

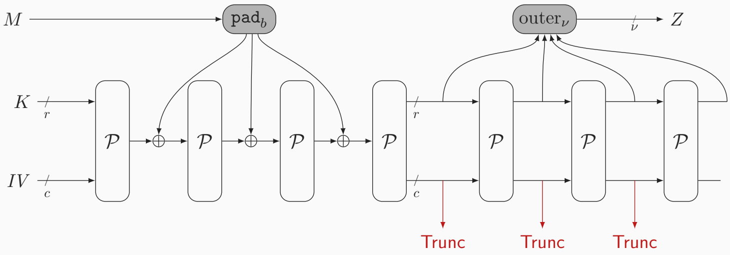
$$\mathbf{Adv}^{\mu\text{-PRF}}_{\mathsf{FSKS}^*} = \tilde{\mathcal{O}}\left(\frac{\mathcal{N}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^b} + \frac{\mathcal{N}}{2^c}\right)$$

MacaKey



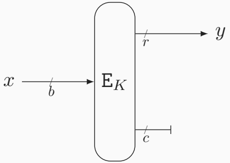
- We have: full-state absorption, multi-user security optimized, tight security bounds

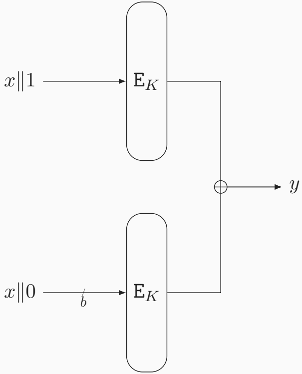
The Full-State Keyed Sponge



- We have: full-state absorption, multi-user security optimized, tight security bounds
- Can we get full-state squeezing? Those inner parts seem wasted..

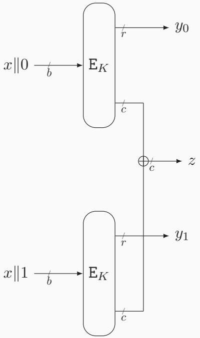
Truncation and Summation





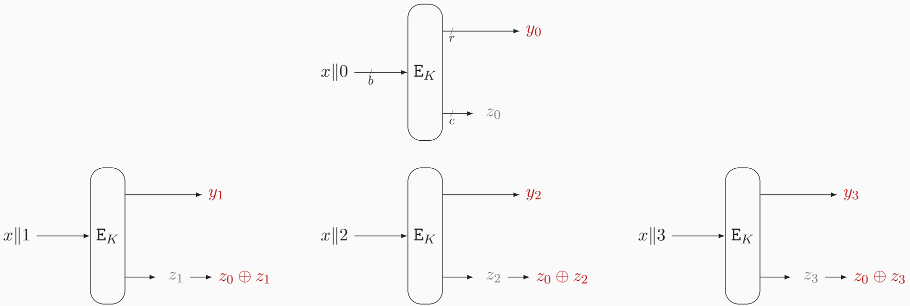
- Holds in the secret permutation setting (i.e., block cipher-based)
- Truncation and Summation: common PRP-to-PRF conversions

The Summation Truncation Hybrid (STH) [GM20] (1/2)



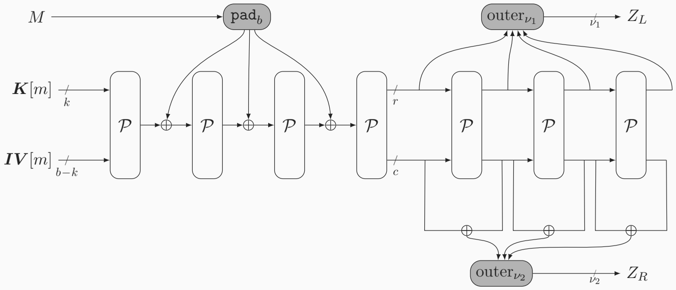
- With truncation, discarding truncated parts is wasteful
- Can group the evaluations two by two, sum them together $\implies$ get c bits for free, *without sacrificing security*

The Summation Truncation Hybrid (STH) [GM20] (2/12)



- Can be generalized to larger groups (in a CENC [Iwa06] fashion)
- Group of w evaluations: output of $r + b(w-1)$ bits $\implies$ approaches b bit per permutation call when w is large

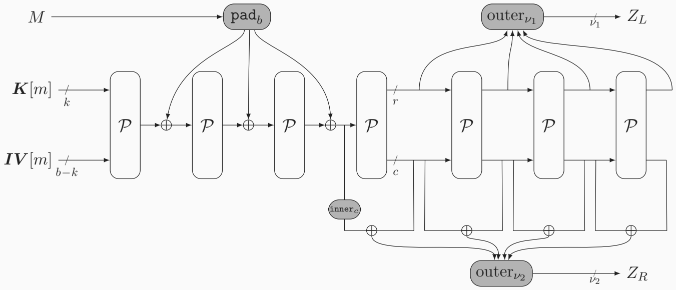
Incorporating the STH into the FSKS: MacaKey



- V1: First call during squeezing phase: r bits of output

  After the first call: b bits of output per permutation call

Incorporating the STH into the FSKS: MacaKey



- V1: First call during squeezing phase: r bits of output

  After the first call: b bits of output per permutation call
- V2: b bits squeezed per permutation call

- We prove

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{MacaKey}}(\mathcal{A}) = \tilde{\mathcal{O}}\left(\frac{\mathcal{N} \cdot u^{\boldsymbol{IV}}_{\max}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^b} + \frac{\mathcal{M}^2}{2^c} + \frac{(\mathcal{L}+1)\mathcal{N}}{2^c}\right)$$

- $u^{\boldsymbol{IV}}_{\max}$ depends on the choice of the IVs $(1 \leq u^{\boldsymbol{IV}}_{\max} \leq \mu)$
- $\mathcal{L}$ denotes the number of states where the adversary has control on the outer part

- We prove

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{MacaKey}}(\mathcal{A}) = \tilde{\mathcal{O}}\left(\frac{\mathcal{N}\cdot u^{\boldsymbol{IV}}_{\max}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^b} + \frac{\mathcal{M}^2}{2^c} + \frac{(\mathcal{L}+1)\mathcal{N}}{2^c}\right)$$

- $u^{\boldsymbol{IV}}_{\max}$ depends on the choice of the IVs $(1 \leq u^{\boldsymbol{IV}}_{\max} \leq \mu)$
- $\mathcal{L}$ denotes the number of states where the adversary has control on the outer part
- **Concrete example:** with $b = 320,\ r = 128,\ c = 192,\ k = 128, \mu \ll 2^{20}$, one IV per user, $\mathcal{L} = 0$ and assuming $\mathcal{M} \ll 2^{64}\mu$, this gives 128 bits of security

Conclusions

- **Hashing improvements:** feed-forward mechanisms enhance (second) preimage security:
  - SPONGE-DM with ideal preimage resistance ($\approx 2^{\nu}$) and significantly improved second preimage bound,
  - SPONGE-EDM^{a} that gives a tunable trade-off
- **PRF improvements:** STH into FSKS allows full-state squeezing without sacrificing generic security
- **AEAD side:** not discussed, design space larger, a lot of possible optimizations

- **Hashing improvements:** feed-forward mechanisms enhance (second) preimage security:
  - SPONGE-DM with ideal preimage resistance ($\approx 2^{\nu}$) and significantly improved second preimage bound,
  - SPONGE-EDM^{a} that gives a tunable trade-off
- **PRF improvements:** STH into FSKS allows full-state squeezing without sacrificing generic security
- **AEAD side:** not discussed, design space larger, a lot of possible optimizations

Thank you for your attention!

Elena Andreeva, Joan Daemen, Bart Mennink, and Gilles Van Assche.
Security of Keyed Sponge Constructions Using a Modular Proof Approach.
In Gregor Leander, 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, 2015.

Elena Andreeva, Bart Mennink, and Bart Preneel.
Security Reductions of the Second Round SHA-3 Candidates.
In Mike Burmester, Gene Tsudik, Spyros S. Magliveras, and Ivana Ilic, editors, *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, volume 6531 of *Lecture Notes in Computer Science*, pages 39–53. Springer, 2010.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Sponge Functions.
Ecrypt Hash Workshop 2007, May 2007.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
On the Indifferentiability of the Sponge Construction.
In Nigel P. Smart, 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, 2008.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Cryptographic sponge functions, 2011.
https://keccak.team/files/CSF-0.1.pdf.

- Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
  On the Security of the Keyed Sponge Construction.
  Symmetric Key Encryption Workshop (SKEW 2011), 2011.
- Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
  Permutation-based encryption, authentication and authenticated encryption.
  Directions in Authenticated Ciphers, July 2012.
- Donghoon Chang, Morris Dworkin, Seokhie Hong, John Kelsey, and Mridul Nandi.
  A Keyed Sponge Construction with Pseudorandomness in the Standard Model.
  NIST SHA–3 Workshop, March 2012.

Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya.
Merkle-Damgård Revisited: How to Construct a Hash Function.
In Victor Shoup, 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, 2005.

Christoph Dobraunig and Bart Mennink.
Security of the Suffix Keyed Sponge.
IACR Trans. Symmetric Cryptol., 2019(4):223–248, 2019.

Christoph Dobraunig and Bart Mennink.
Tightness of the Suffix Keyed Sponge Bound.
IACR Trans. Symmetric Cryptol., 2020(4):195–212, 2020.

Christoph Dobraunig and Bart Mennink.
Generalized Initialization of the Duplex Construction.
In Christina Pöpper and Lejla Batina, editors, *Applied Cryptography and Network Security - 22nd International Conference, ACNS 2024, Abu Dhabi, United Arab Emirates, March 5-8, 2024, Proceedings, Part II*, volume 14584 of *Lecture Notes in Computer Science*, pages 460–484. Springer, 2024.

Joan Daemen, Bart Mennink, and Gilles Van Assche.
Full-State Keyed Duplex with Built-In Multi-user Support.
In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017,*

Proceedings, Part II, volume 10625 of *Lecture Notes in Computer Science*, pages 606–637. Springer, 2017.

Robin Foekens.
Security of the Sponge Construction with a Random Transformation.
Bachelor's Thesis, 2023.

Chun Guo, Kai Hu, Yanhong Fan, Yong Fu, and Meiqin Wang.
Adding Feeding Forward Back to the Sponge Construction.
IACR Cryptol. ePrint Arch., page 1006, 2025.

Aldo Gunsing and Bart Mennink.
The Summation-Truncation Hybrid: Reusing Discarded Bits for Free.
In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 187–217. Springer, 2020.

Jian Guo, Thomas Peyrin, and Axel Poschmann.
The PHOTON Family of Lightweight Hash Functions.
In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.

Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro.
The Exact PRF Security of Truncation: Tight Bounds for Keyed Sponges and Truncated CBC.
In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 368–387. Springer, 2015.

Tetsu Iwata.
New Blockcipher Modes of Operation with Beyond the Birthday Bound Security.
In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006.

Charlotte Lefevre and Bart Mennink.
Tight Preimage Resistance of the Sponge Construction.
In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 185–204. Springer, 2022.

Charlotte Lefevre and Bart Mennink.
Permutation-Based Hashing Beyond the Birthday Bound.
IACR Trans. Symmetric Cryptol., 2024(1):71–113, 2024.

Bart Mennink.
Key Prediction Security of Keyed Sponges.
IACR Trans. Symmetric Cryptol., 2018(4):128–149, 2018.

Bart Mennink.
Understanding the Duplex and Its Security.
IACR Trans. Symmetric Cryptol., 2023(2):1–46, 2023.

Ueli M. Maurer, Renato Renner, and Clemens Holenstein.
Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology.
In Moni Naor, 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, 2004.

Bart Mennink, Reza Reyhanitabar, and Damian Vizár.
Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption.
In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 465–489. Springer, 2015.

Yusuke Naito.
Sandwich Construction for Keyed Sponges: Independence Between Capacity and Online Queries.
In Sara Foresti and Giuseppe Persiano, editors, *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, volume 10052 of *Lecture Notes in Computer Science*, pages 245–261, 2016.

- Yusuke Naito and Kazuo Ohta.
  Improved Indifferentiable Security Analysis of PHOTON.
  In Michel Abdalla and Roberto De Prisco, 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, 2014.
- Yusuke Naito and Kan Yasuda.
  New Bounds for Keyed Sponges with Extendable Output: Independence Between Capacity and Message Length.
  In Thomas Peyrin, 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, 2016.

Phillip Rogaway and Thomas Shrimpton.
Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance.
In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.