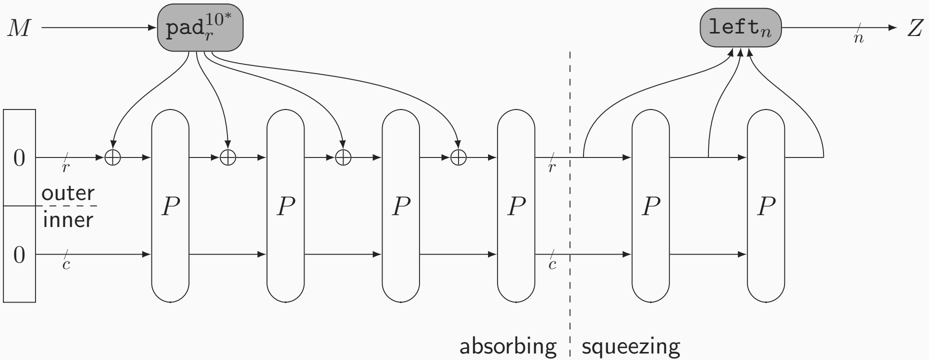
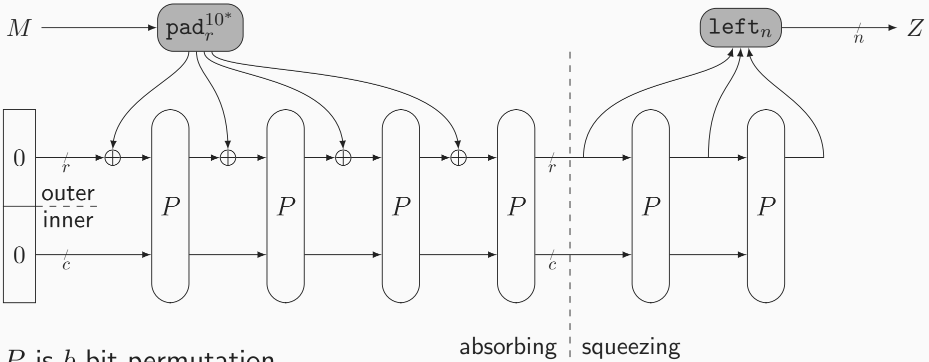
Radbound University

To Pad or Not to Pad? Padding-Free Arithmetization-Oriented Sponges

Charlotte Lefevre, Mario Marhuenda Beltrán, Bart Mennink
Radboud University (The Netherlands)
FSE 2025, Rome
March 17, 2025

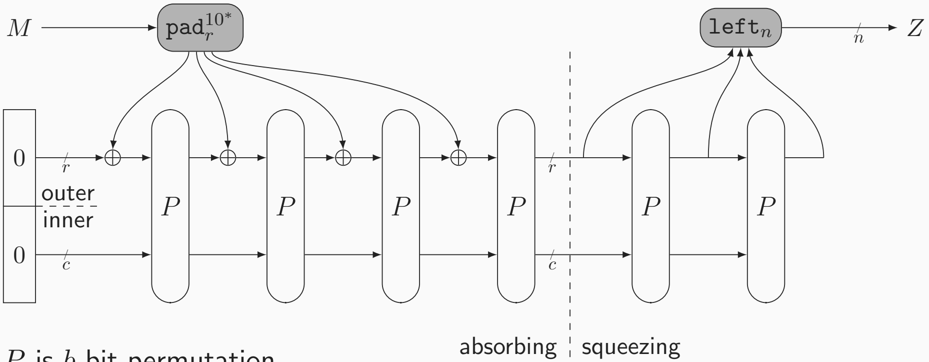
Sponge Construction [BDPV07]





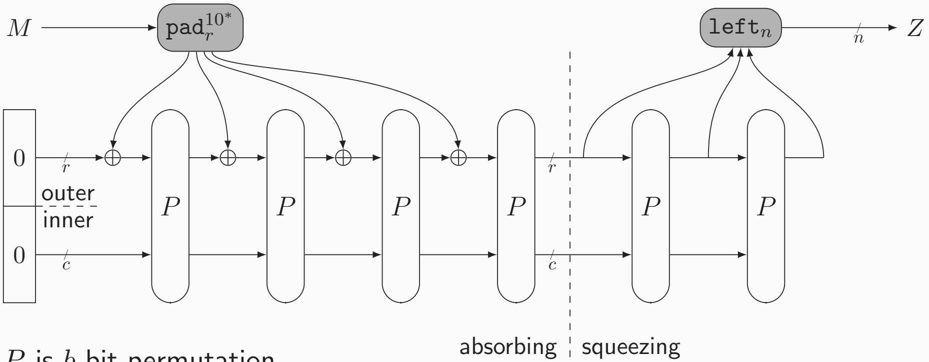
- P is b-bit permutation

Sponge Construction [BDPV07]



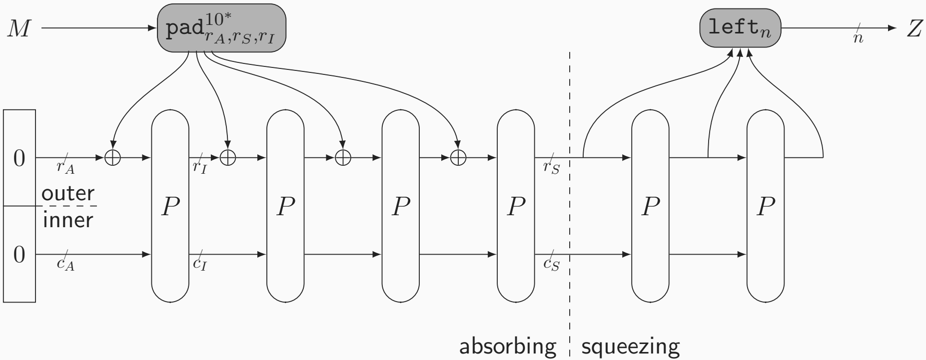
- P is b-bit permutation
  - r is the rate
  - c is the capacity
  - $b = r + c$

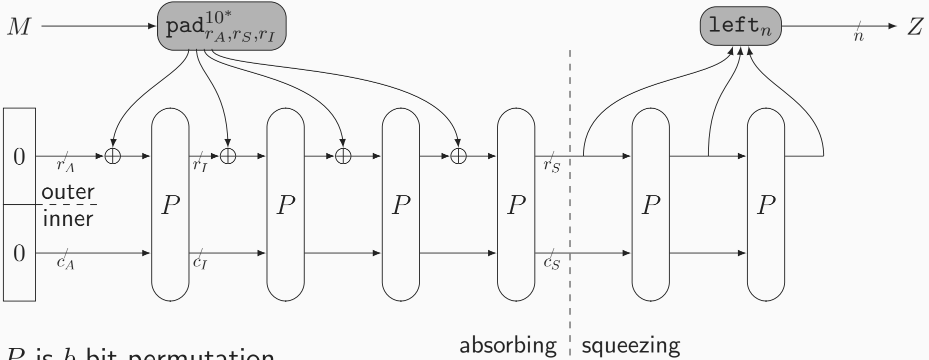
Sponge Construction [BDPV07]



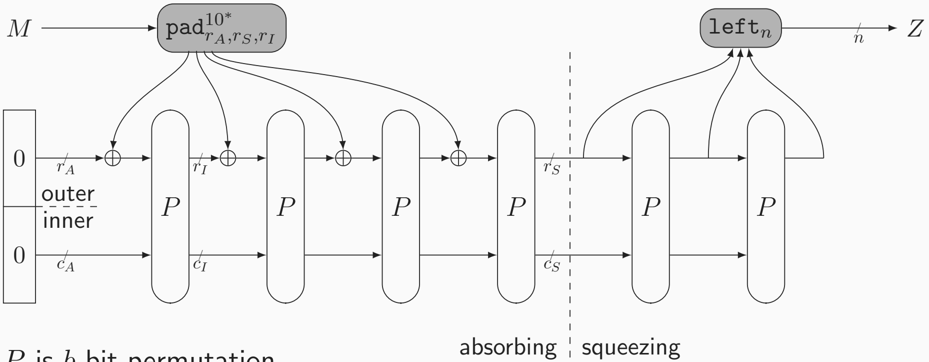
- P is b-bit permutation
  - r is the rate
  - c is the capacity
  - $b = r + c$
- **Security**: generically behaves like $\mathtt{RO}$ up to $\mathcal{O}\left(2^{c/2}\right)$ queries [BDPV08]

Generalized Sponge Construction [NO14]

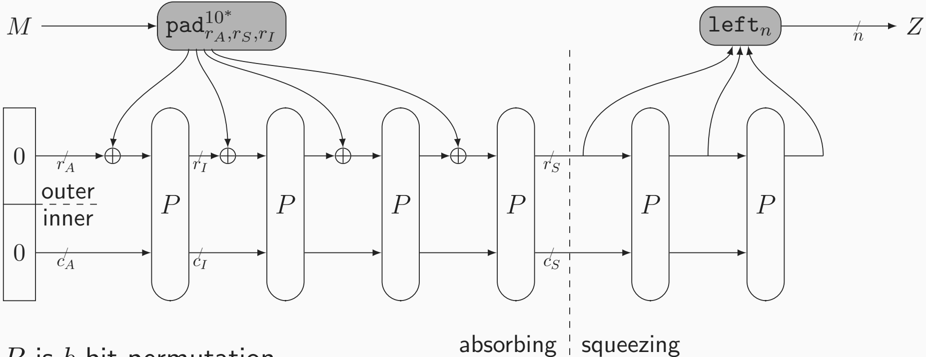




- P is b-bit permutation

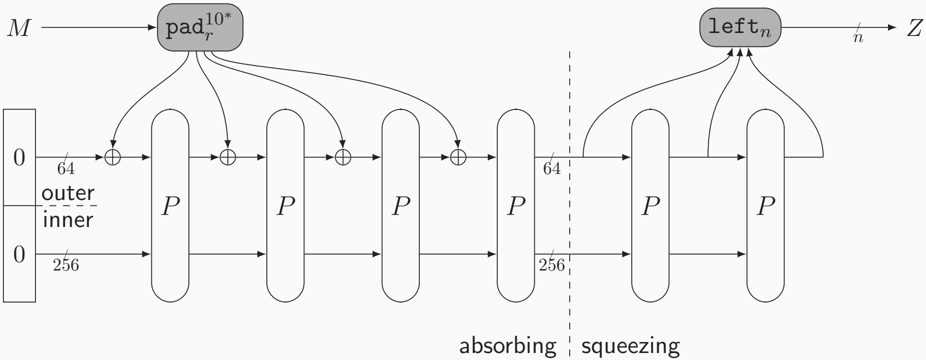


- P is b-bit permutation
  - r_A, r_I, r_S are the rates and usually $r_A \simeq r_I + c_I/2$
  - c_A, c_I, c_S are the capacities
  - $b = r_A + c_A = r_I + c_I = r_S + c_S$

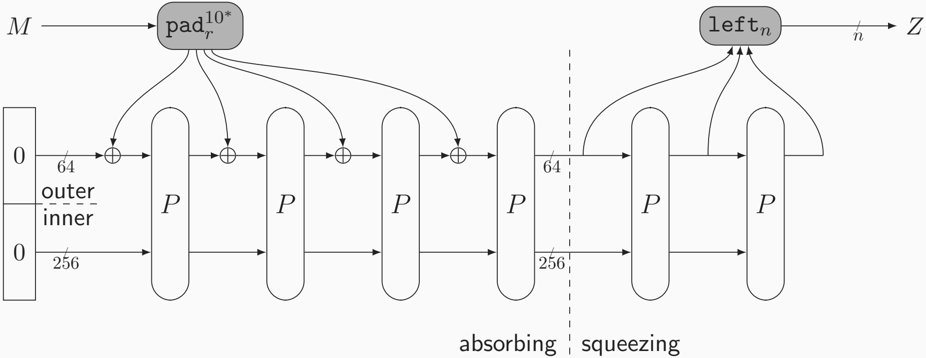


- P is b-bit permutation
  - r_A, r_I, r_S are the rates and usually $r_A \simeq r_I + c_I/2$
  - c_A, c_I, c_S are the capacities
  - $b = r_A + c_A = r_I + c_I = r_S + c_S$
- **Security**: behaves like RO up to $\mathcal{O}\left(2^{\min\{c_I/2, c_S/2\}}\right)$ queries [NO14]

A Concrete Example

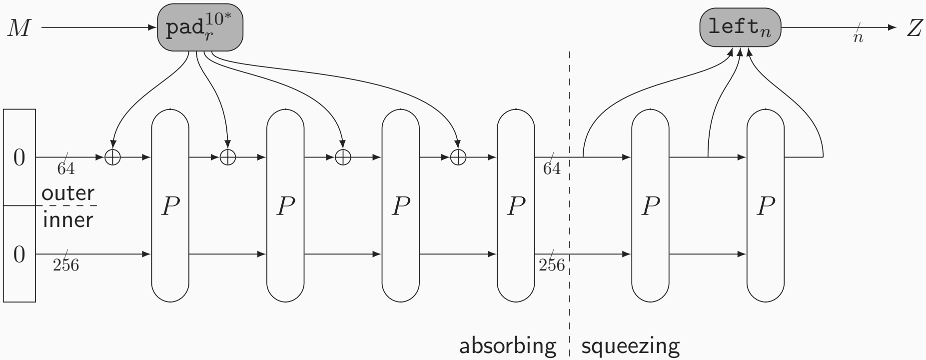


A Concrete Example



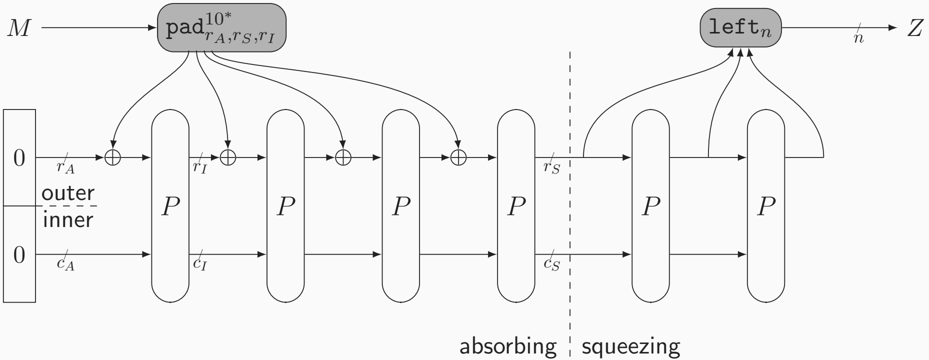
- Ascon-Hash256: $b = 320, r = 64, c = 256$

A Concrete Example



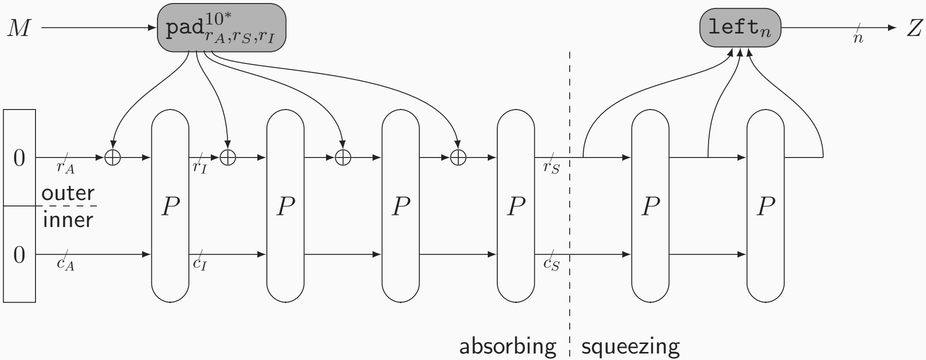
- Ascon-Hash256: $b = 320, r = 64, c = 256$
- Padding cost: small overhead

A Reinforced Concrete Example [GKL+22]



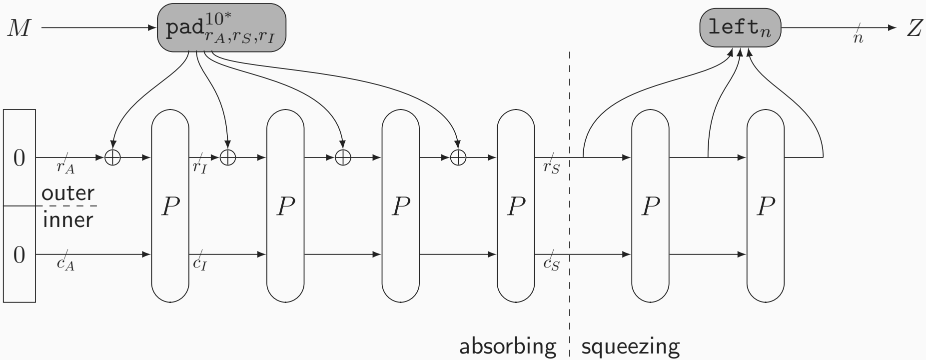
- Security level: 128

A Reinforced Concrete Example [GKL+22]



- Security level: 128
- $p \simeq 2^{256}$

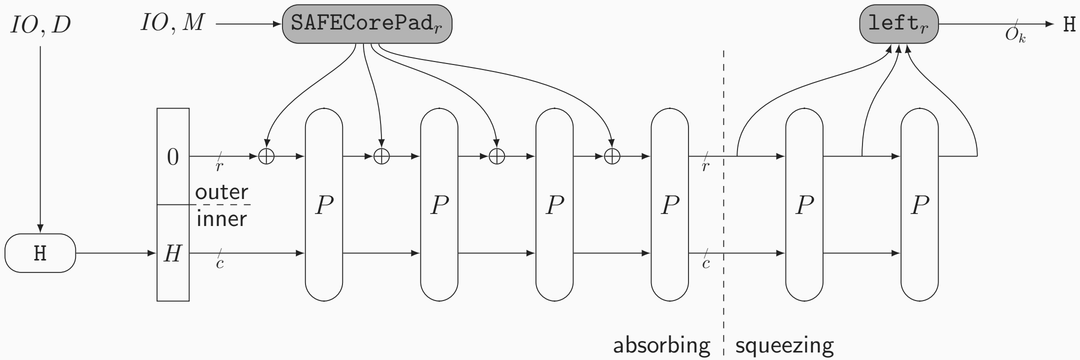
A Reinforced Concrete Example [GKL+22]

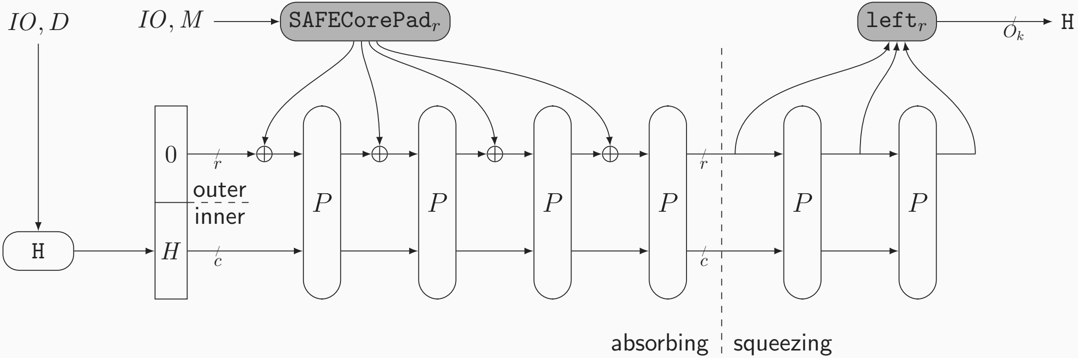


- Security level: 128
- $p \simeq 2^{256}$
- $b = 3$, $r = 2$, and $c = 1$

Padding may cost 256 bits!

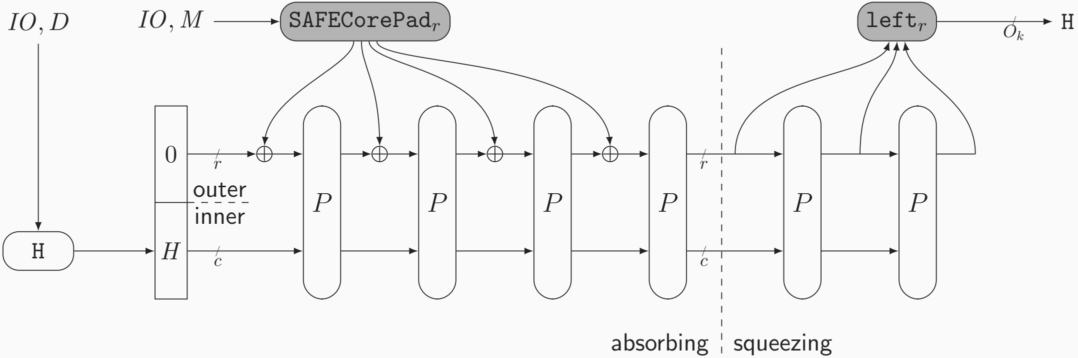
The Rise Of SAFE [AKQ22]: A Restricted Setting Solution





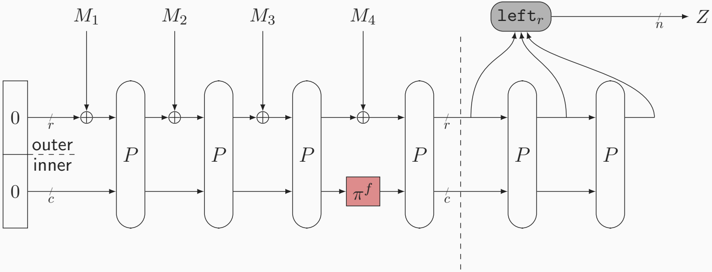
- **Security**: behaves like RO up to $\mathcal{O}\left(2^{c/2}\right)$ queries [KMM23]

The Rise Of SAFE [AKQ22]: A Restricted Setting Solution

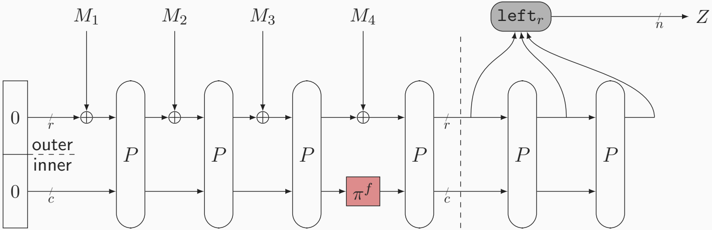


- **Security**: behaves like RO up to $\mathcal{O}(2^{c/2})$ queries [KMM23]
- Inflexible message sizes

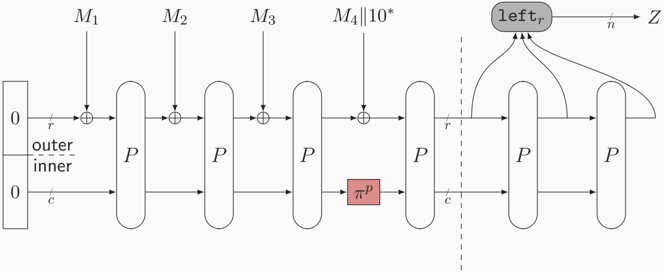
- Message length is a multiple of r

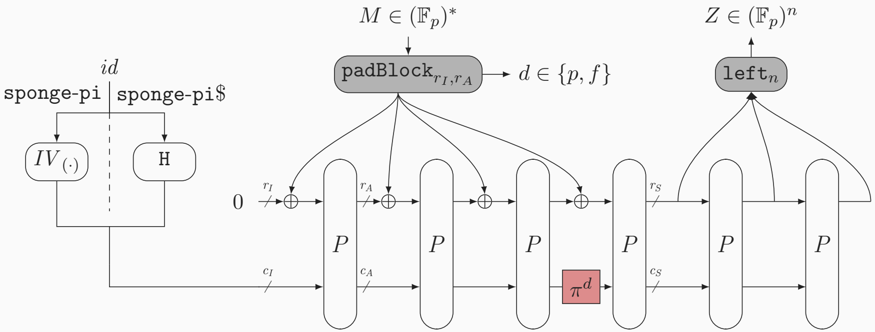


- Message length is a multiple of r

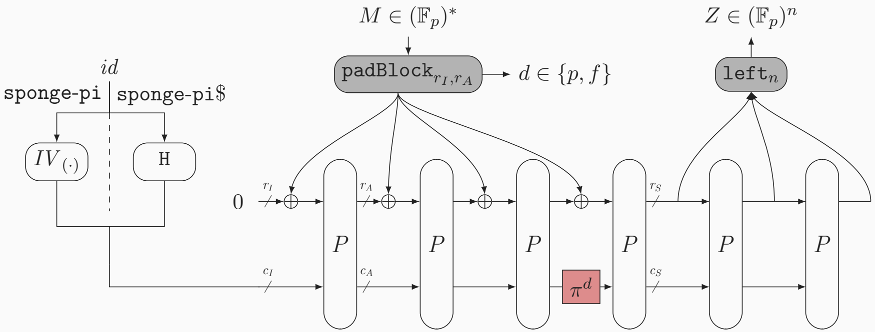


- Message length is not a multiple of r

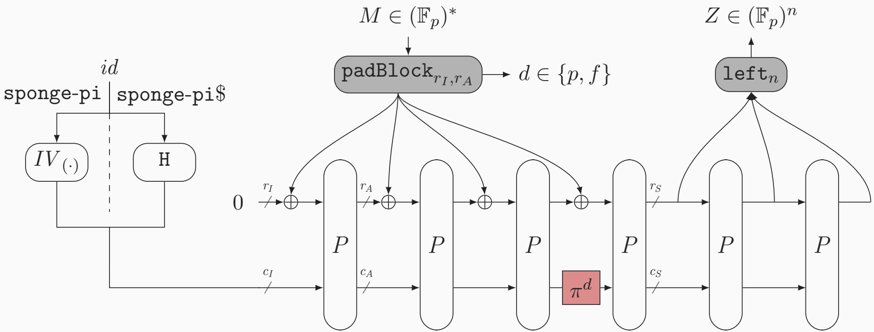




id
sponge-pi
sponge-pi$
$IV_{(\cdot)}$
H
$M \in (\mathbb{F}_p)^*$
$\texttt{padBlock}_{r_I, r_A}$
$d \in \{p, f\}$
$Z \in (\mathbb{F}_p)^n$
$\texttt{left}_n$
0
r_I
r_A
r_S
c_I
c_A
c_S
P
π^d

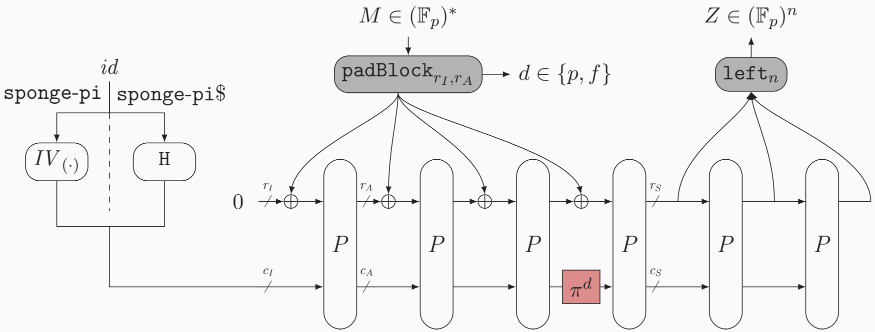


- sponge-pi: User-determined IVs



- sponge-pi: User-determined IVs
- sponge-pi$: Randomly determined IVs

Our Constructions: sponge-pi And sponge-pi$



- sponge-pi: User-determined IVs
- sponge-pi$: Randomly determined IVs

Same, but different

sponge-pi and sponge-pi$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

sponge-pi and sponge-pi$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

sponge	$\dfrac{Q_P^2}{p^c} + \dfrac{\mu Q_P}{p^c}$
`sponge-pi`	$\dfrac{7Q_P^2}{p^c} + \dfrac{7\mu Q_P}{2p^c}$
`sponge-pi$`	$\dfrac{7Q_P^2}{p^c} + \dfrac{22 Q_P Q_H}{p^c}$

sponge-pi and sponge-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to H

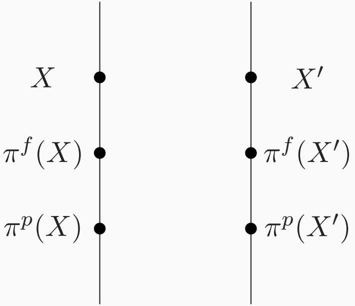
sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
sponge-pi	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
sponge-pi\$	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

Where does the 7 come from?

sponge-pi and sponge-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to H

sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
sponge-pi	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
sponge-pi\$	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

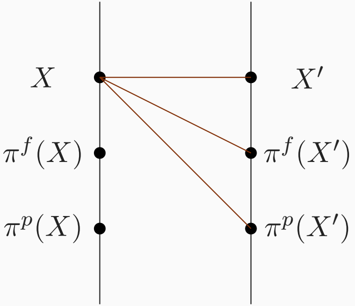


Where does the 7 come from?

sponge-pi and sponge-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
`sponge-pi`	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
`sponge-pi$`	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

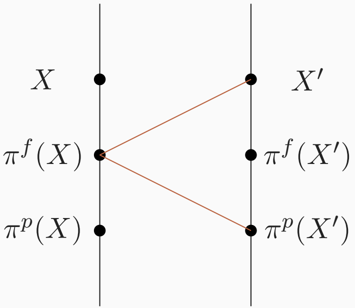


Where does the 7 come from?

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
`sponge-pi`	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
`sponge-pi$`	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

Where does the 7 come from?



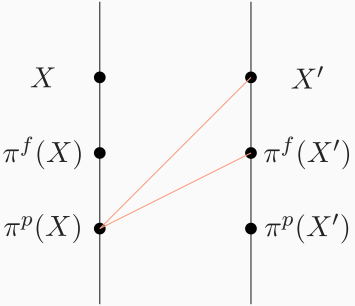
- Horizontal lines are equivalent

sponge-pi and sponge-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to H

sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
sponge-pi	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
sponge-pi\$	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

Where does the 7 come from?

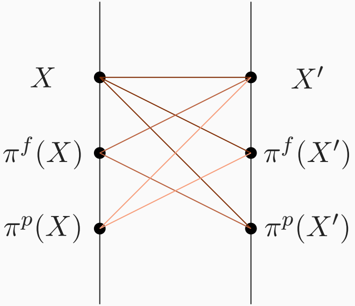


- Horizontal lines are equivalent

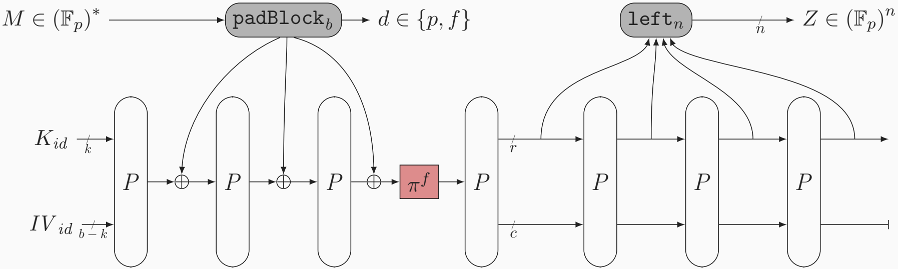
- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

sponge	$\frac{Q_P^2}{p^c} + \frac{\mu Q_P}{p^c}$
`sponge-pi`	$\frac{7Q_P^2}{p^c} + \frac{7\mu Q_P}{2p^c}$
`sponge-pi$`	$\frac{7Q_P^2}{p^c} + \frac{22 Q_P Q_H}{p^c}$

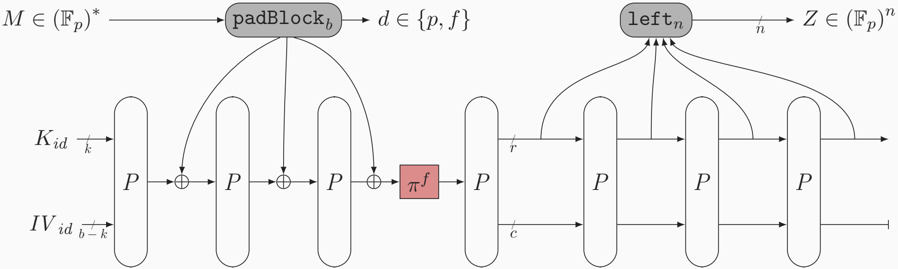
Where does the 7 come from?



- Horizontal lines are equivalent

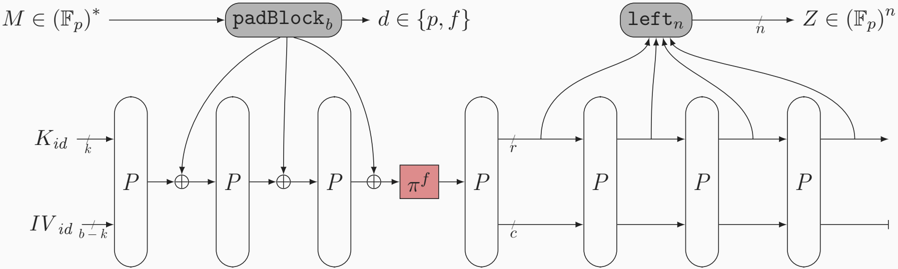


$M \in (\mathbb{F}_p)^*$
padBlock$_b$
$d \in \{p, f\}$
left$_n$
n
$Z \in (\mathbb{F}_p)^n$
K_{id}
k
IV_{id}
$b-k$
P
P
P
π^f
P
r
c
P
P
P

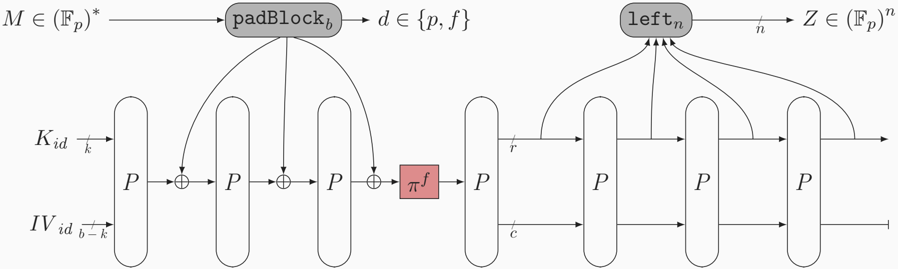


- P is b-bit permutation

Our Constructions: fks-pi

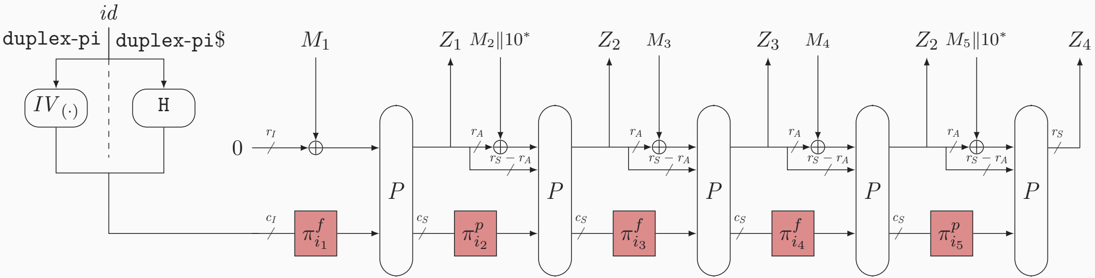


- P is b-bit permutation
  - k is the key size
  - c is the capacity
  - $b = r + c$



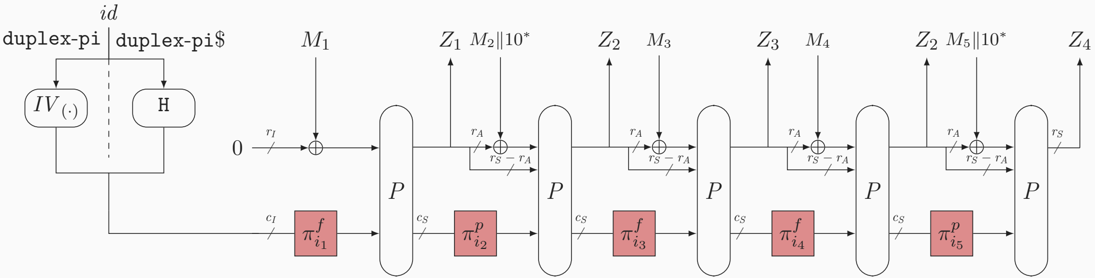
- P is b-bit permutation
  - k is the key size
  - c is the capacity
  - $b = r + c$
- **Security**: behaves like RO up to $\mathcal{O}\big(\min\{p^k, p^b/\sigma, p^c\}\big)$, where σ is the online complexity

Our Constructions: duplex-pi and duplex-pi$

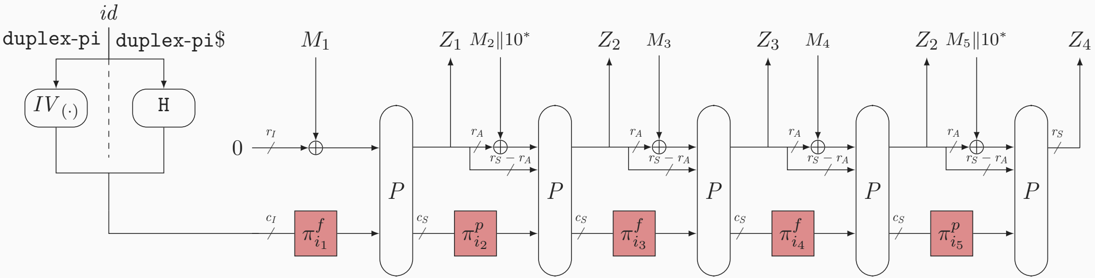


- ξ: Number of different NCPs

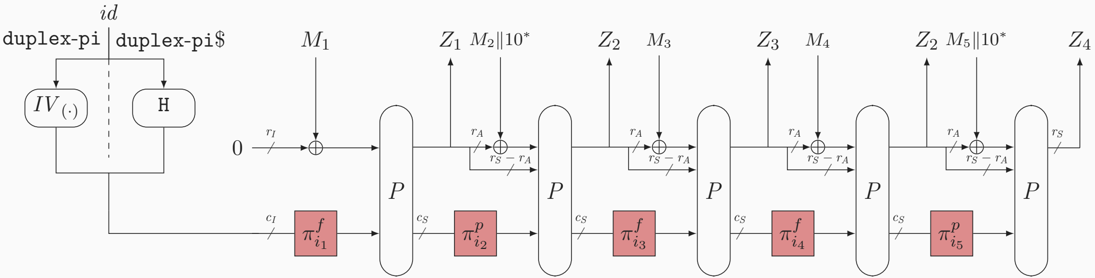
Our Constructions: duplex-pi and duplex-pi\$



- ξ: Number of different NCPs
- `duplex-pi`: User-determined IVs



- ξ: Number of different NCPs
- duplex-pi: User-determined IVs
- duplex-pi$: Randomly determined IVs



- ξ: Number of different NCPs
- `duplex-pi`: User-determined IVs
- `duplex-pi$`: Randomly determined IVs

Same, but different

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

duplex-pi and duplex-pi$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

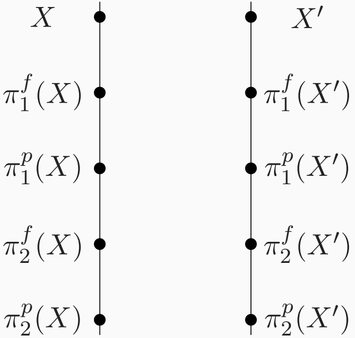
Where does the ξ^2 come from?

duplex-pi and duplex-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

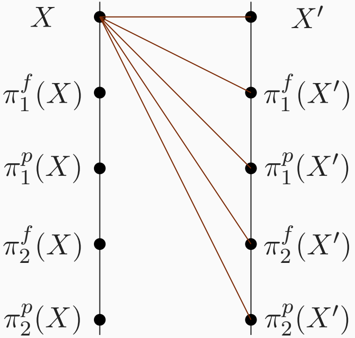
Where does the ξ^2 come from?



- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?

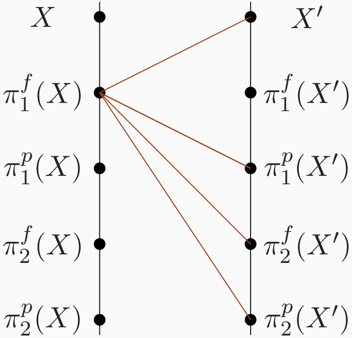


duplex-pi and duplex-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?



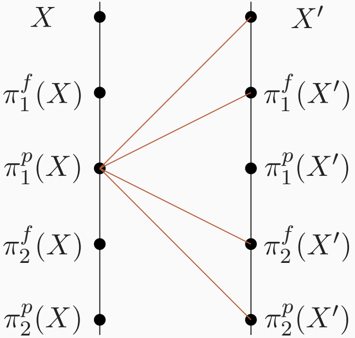
- Horizontal lines are equivalent

duplex-pi and duplex-pi$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?



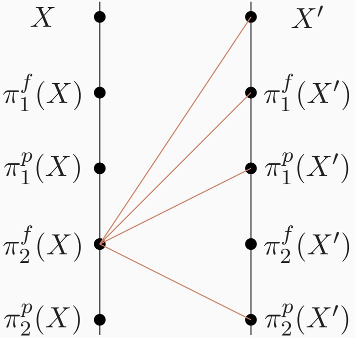
- Horizontal lines are equivalent

duplex-pi and duplex-pi\$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?



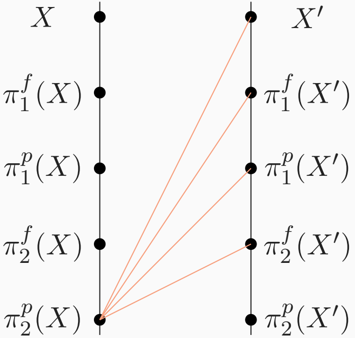
- Horizontal lines are equivalent

duplex-pi and duplex-pi$: Simplified Security Comparison

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to `H`

FKD	$\frac{Q_P^2}{p^c} + \frac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\frac{\xi^2 Q_P^2}{2p^c} + \frac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\frac{\xi^2 Q_P^2}{p^c} + \frac{\xi^2 Q_H^2}{p^c} + \frac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?

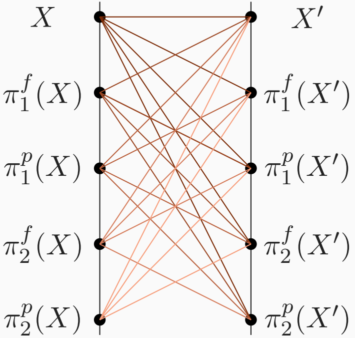


- Horizontal lines are equivalent

- Simplified setting: $c_I = c_A = c_S = c$
- Q_P: Queries to P
- Q_H: Queries to $\mathtt{H}$

FKD	$\dfrac{Q_P^2}{p^c} + \dfrac{2\nu_{r,c}^{2Q_P} Q_P}{p^c}$
`duplex-pi`	$\dfrac{\xi^2 Q_P^2}{2p^c} + \dfrac{\xi^2 \mu Q_P}{p^c}$
`duplex-pi$`	$\dfrac{\xi^2 Q_P^2}{p^c} + \dfrac{\xi^2 Q_H^2}{p^c} + \dfrac{\xi^2 Q_H Q_P}{p^c}$

Where does the ξ^2 come from?



- Horizontal lines are equivalent
- Total: $\underbrace{\xi^2}_{\text{pairs}} - \underbrace{(\xi - 1)}_{\text{horizontal}}$

Conclusion

- Introduction and formal generic analysis of `sponge-pi`($\$$), `duplex-pi`($\$$), and `fks-pi`

Conclusion

- Introduction and formal generic analysis of `sponge-pi`($\$$), `duplex-pi`($\$$), and `fks-pi`
- Generic security bound is **the same** as previous constructions

Conclusion

- Introduction and formal generic analysis of `sponge-pi`($\$$), `duplex-pi`($\$$), and `fks-pi`
- Generic security bound is **the same** as previous constructions
  - Small (constant) loss in the binary setting

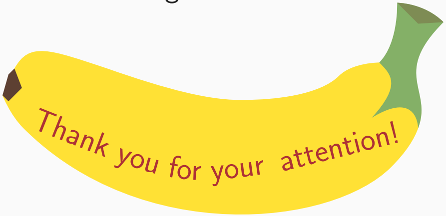
Conclusion

- Introduction and formal generic analysis of `sponge-pi`($), `duplex-pi`($), and `fks-pi`
- Generic security bound is **the same** as previous constructions
  - Small (constant) loss in the binary setting
  - Padding-free hashing and duplexing

Conclusion

- Introduction and formal generic analysis of `sponge-pi`($\$$), `duplex-pi`($\$$), and `fks-pi`
- Generic security bound is **the same** as previous constructions
  - Small (constant) loss in the binary setting
  - Padding-free hashing and duplexing
  - Allows for more efficient hashing in finite fields

- Introduction and formal generic analysis of `sponge-pi`($), `duplex-pi`($), and `fks-pi`
- Generic security bound is **the same** as previous constructions
  - Small (constant) loss in the binary setting
  - Padding-free hashing and duplexing
  - Allows for more efficient hashing in finite fields



- Jean-Philippe Aumasson, Dmitry Khovratovich, and Porçu Quine.
  SAFE (Sponge API for Field Elements) – A Toolbox for ZK Hash Applications, 2022.
- Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
  Sponge Functions, 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.

- Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch.
  Reinforced Concrete: A Fast Hash Function for Verifiable Computation.
  In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1323–1335, New York, NY, USA, 2022. Association for Computing Machinery.

- Dmitry Khovratovich, Mario Marhuenda Beltrán, and Bart Mennink.
  Generic Security of the SAFE API and Its Applications.
  In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VIII*, volume 14445 of *Lecture Notes in Computer Science*, pages 301–327. Springer, 2023.

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.