Introduction to Provable Security for Symmetric-Key Cryptography

Case Study: The Sponge Construction

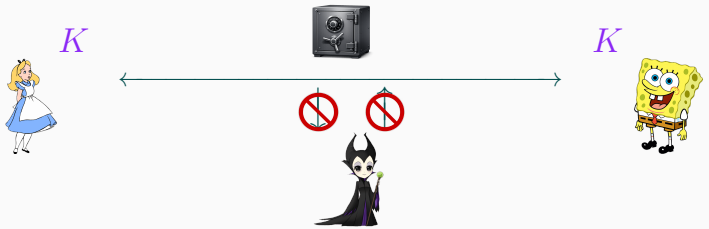
Charlotte Lefevre
(many slides from joint works with Bart Mennink)

SAC 2026 Summer School

Overview

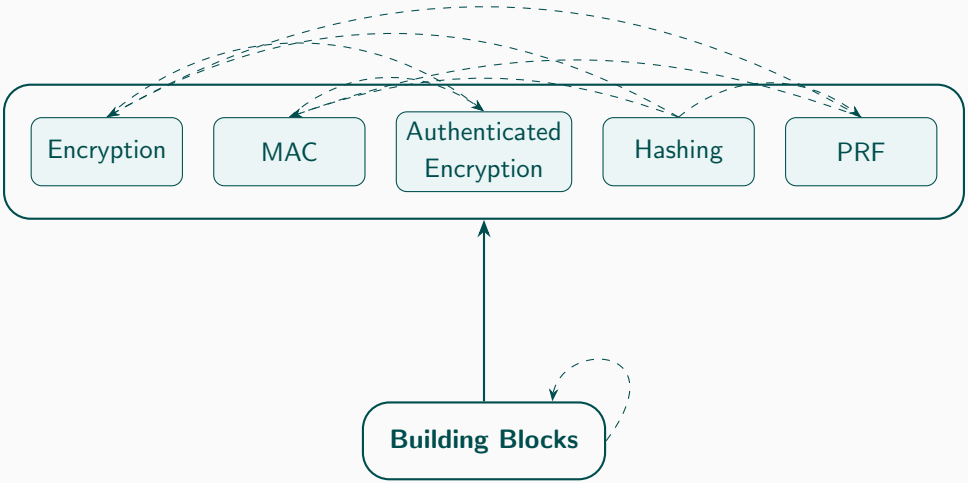
Foundations

Symmetric-Key: Authentication and Encryption



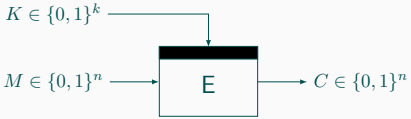
- Two parties, Alice and Bob, want to communicate over a public channel
  - They have agreed on a secret key K, used to protect their messages
- Eve may try to eavesdrop, tamper with, or disrupt this communication
- This leads to two core security goals:
  - Confidentiality: Eve learns nothing about the message contents
  - Authenticity: Manipulation/replay/delay of messages is detected

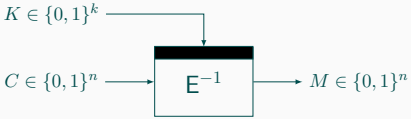
Symmetric-Key Algorithms (Oversimplified)



Dedicated designs also exist (e.g., some stream ciphers)

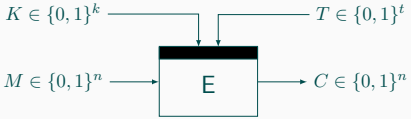
Block Cipher

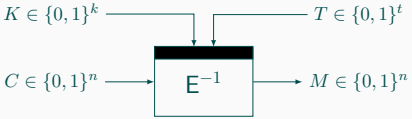




- Using key K, message M is bijectively transformed to ciphertext C
- Example: AES with $k \in \{128, 192, 256\}$; $n = 128$

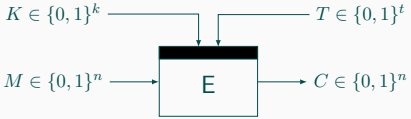
Block Cipher

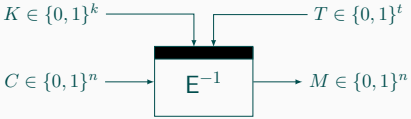




- Using key K, message M is bijectively transformed to ciphertext C
- Example: AES with $k \in \{128, 192, 256\}$; $n = 128$
- Generalization: tweakable block cipher

Block Cipher

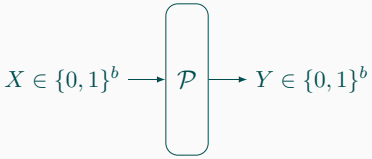




- Using key K, message M is bijectively transformed to ciphertext C
- Example: AES with $k \in \{128, 192, 256\}$; $n = 128$
- Generalization: tweakable block cipher

Permutation

- A bijective function (no key)
- Example: Ascon ($b = 320$), Keccak-f[1600]



Foundations

Security Models With Examples

The Big Picture (1/2)

- We cannot prove the security of most cryptosystems directly
  (No full security proof for SHA-2, SHA-3, AES-GCM, ...)
- In practice, we rely on two complementary approaches:
  (1) **Cryptanalysis:** extensively attack the building block and/or the algorithm
  (2) **Provable Security:** prove security under some assumptions on the building block/adversary

The Big Picture (2/2)

Provable Security:

- **Standard model:** "If B is secure, then mode$[B]$ is secure", or

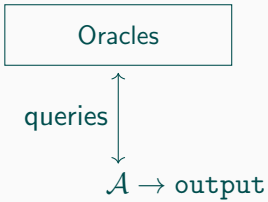
$$\mathbf{Adv}^{\text{sec-model}}_{\text{mode}[B]}(\mathcal{A}) \leq \text{some terms} + \mathbf{Adv}^{\text{other-sec-model}}_{B}(\mathcal{A}')$$

- **Ideal model:** replaces B with an ideal object (e.g. random oracle):

$$\mathbf{Adv}^{\text{sec-model}}_{\text{mode}[\text{ideal-}B]}(\mathcal{A}) \leq \text{some terms (do not depend on } B)$$

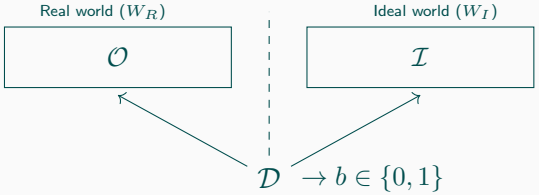
$\hookrightarrow$ Allows to prove more things, but **interpret results carefully**

Security Games We'll Look at



- $\mathcal{A}$ wins if $\texttt{output}$ satisfies a set of conditions
- **Information-theoretic adversary**: $\mathcal{A}$'s ressources are only measured by its number of oracle queries

Indistinguishability



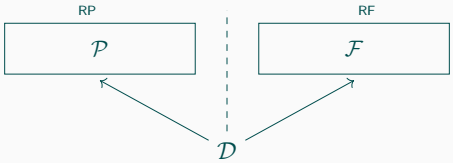
$b' = 0$ if real world, else $b' = 1$

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ind}}(\mathcal{D}) &= \left|2\mathbf{Pr}\left(b = b'\right) - 1\right| \\ &= \left|\mathbf{Pr}\left(\mathcal{D}^{\mathcal{O}} \to 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathcal{I}} \to 1\right)\right|\end{aligned}$$

Goal: bound $\mathbf{Adv}^{\mathrm{ind}}(\mathcal{D})$ as a function of $\mathcal{D}$'s ressources

Example: PRP/PRF Switching Lemma (1/2)

Happens frequently as an intermediate step in security proofs



- $\mathcal{P} : \{0,1\}^n \to \{0,1\}^n$ is a random permutation (forward queries only)
- $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ is a random function
- The adversary makes at most q fresh queries
- Proof using the fundamental lemma of game-playing [BR06]

Example: PRP/PRF Switching Lemma (2/2)

On a fresh input x,

RP:

$y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RF:

$y \xleftarrow{\$} \{0,1\}^n$

$T \leftarrow T \cup \{y\}$

return y

- T is initialized as the empty set

Example: PRP/PRF Switching Lemma (2/2)

On a fresh input x,

RP:

$y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RP':

$y \xleftarrow{\$} \{0,1\}^n$

If $y \in T$

— set **bad**

— $y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RF:

$y \xleftarrow{\$} \{0,1\}^n$

$T \leftarrow T \cup \{y\}$

return y

- T is initialized as the empty set
- Rewrite RP into RP'

Example: PRP/PRF Switching Lemma (2/2)

On a fresh input x,

RP:

$y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RP':

$y \xleftarrow{\$} \{0,1\}^n$

If $y \in T$

— set **bad**

— $y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RF':

$y \xleftarrow{\$} \{0,1\}^n$

If $y \in T$:

— set **bad**

$T \leftarrow T \cup \{y\}$

return y

RF:

$y \xleftarrow{\$} \{0,1\}^n$

$T \leftarrow T \cup \{y\}$

return y

- T is initialized as the empty set
- Rewrite RP into RP'; RF into RF'

Example: PRP/PRF Switching Lemma (2/2)

On a fresh input x,

RP:

$y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RP':

$y \xleftarrow{\$} \{0,1\}^n$

If $y \in T$

— set **bad**

— $y \xleftarrow{\$} \{0,1\}^n \setminus T$

$T \leftarrow T \cup \{y\}$

return y

RF':

$y \xleftarrow{\$} \{0,1\}^n$

If $y \in T$:

— set **bad**

$T \leftarrow T \cup \{y\}$

return y

RF:

$y \xleftarrow{\$} \{0,1\}^n$

$T \leftarrow T \cup \{y\}$

return y

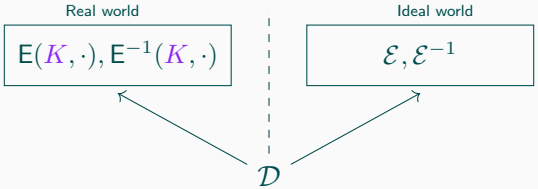
- T is initialized as the empty set
- Rewrite RP into RP'; RF into RF'
- RP' and RF' are identical as long as **bad** is not set

$$\implies \mathbf{Adv}^{\mathsf{RP};\mathsf{RF}}(\mathcal{D}) = \mathbf{Adv}^{\mathsf{RP}';\mathsf{RF}'}(\mathcal{D}) \leq \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RF}'} \text{ sets BAD}\right) \leq \frac{q(q-1)}{2^{n+1}}$$

Foundations

SPRP Security and Even-Mansour

Strong Pseudorandom Permutation (SPRP) Security



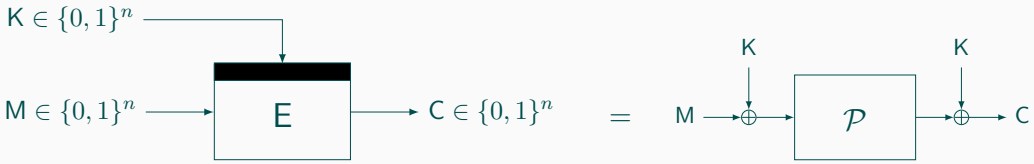
- $\mathsf{E}^{\pm}(K,\cdot)$ for some random secret key K should be indistinguishable from a random permutation $\mathcal{E}^{\pm}$
- $\mathcal{D}$ can make forward or inverse queries[1]
- $\mathsf{E}(\cdot,\cdot)$ is a public algorithm: $\mathcal{D}$ may evaluate $\mathsf{E}^{\pm}$ offline, on any key of its choice

  $\implies$ $\mathcal{D}$'s resources: number of oracle queries and running time

[1] Forward queries only $\implies$ PRP security (weaker notion)

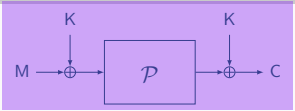
The Even-Mansour Construction [EM91, EM97] (1/4)

The Even-Mansour (EM) construction is a generic way to build a block cipher from a public permutation:

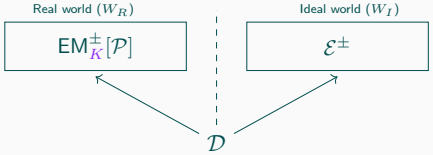


- Encryption: XOR the plaintext with K, apply $\mathcal{P}$, XOR the result with K again
- Decryption: same, using $\mathcal{P}^{-1}$ instead

The Even-Mansour Construction (2/4)



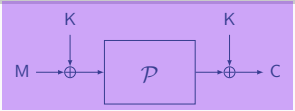
SPRP security:



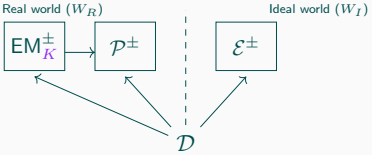
q_E enc/dec queries

- $\mathcal{P}$ is a public permutation

The Even-Mansour Construction (2/4)



SPRP security:

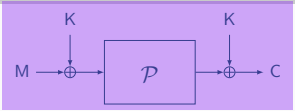


q_E	enc/dec queries
q_P	permutation queries

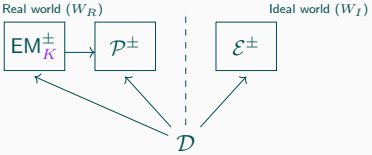
- $\mathcal{P}$ is a public permutation
- Proof treats $\mathcal{P}$ as a uniformly random permutation

  $\Longrightarrow$ Make $\mathcal{P}$ part of the oracles

The Even-Mansour Construction (2/4)



SPRP security:

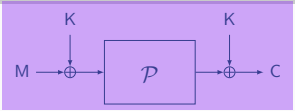


q_E	enc/dec queries
q_P	permutation queries

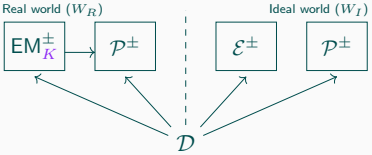
- $\mathcal{P}$ is a public permutation
- Proof treats $\mathcal{P}$ as a uniformly random permutation

  $\Longrightarrow$ Make $\mathcal{P}$ part of the oracles
- This also requires an ideal-world counterpart to compare against

The Even-Mansour Construction (2/4)



SPRP security:

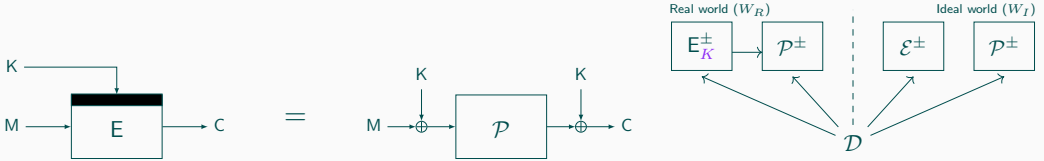


q_E	enc/dec queries
q_P	permutation queries

- $\mathcal{P}$ is a public permutation
- Proof treats $\mathcal{P}$ as a uniformly random permutation

  $\implies$ Make $\mathcal{P}$ part of the oracles
- This also requires an ideal-world counterpart to compare against $\implies$ give access to a random permutation, independent of the block cipher oracle

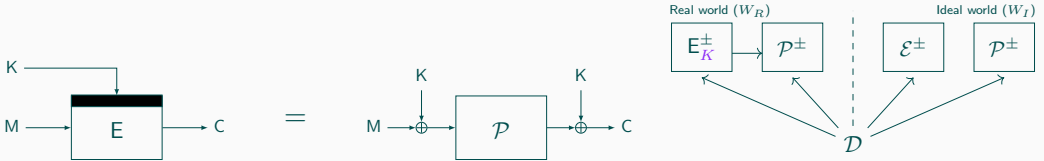
The Even-Mansour Construction (3/4)



- Intuitively, things go wrong when an EM query and a permutation query collide on the same underlying $\mathcal{P}$-call
- Example: an EM query on M internally computes $\mathsf{C} := \mathcal{P}(\mathsf{M} \oplus \mathsf{K}) \oplus \mathsf{K}$. If $\mathcal{D}$ later queries $\mathsf{O} := \mathcal{P}(\mathsf{M} \oplus \mathsf{K})$, then they get $\mathsf{C} \oplus \mathsf{O} = \mathsf{K}$

  $\Longrightarrow$ can be turned into a key recovery attack

The Even-Mansour Construction (4/4)



- As before, security reduces to a **bad event**.[2] Writing $\mathrm{Tab}_{\mathsf{Perm}} = \{(X, Y)\}$ for the table of $\mathcal{P}$-queries so far, **bad** is set when
  - A query $\mathsf{C} \leftarrow \mathsf{E}(\mathsf{M})$ has $\mathsf{M} \oplus \mathsf{K} \in \mathrm{dom}(\mathrm{Tab}_{\mathsf{Perm}})$ or $\mathsf{C} \oplus \mathsf{K} \in \mathrm{rng}(\mathrm{Tab}_{\mathsf{Perm}})$
  - Or same as above, with the roles of E and Perm swapped

$$\implies \mathbf{Adv}^{\mathrm{SPRP}}_{\mathsf{EM}}(\mathcal{D}) \leq \frac{2q_E q_P}{2^n}$$

[2] Shown for PRP security; SPRP is analogous, just more cases

The H-Coefficient Technique [Pat08, CS14]

- We will not use it today, but ubiquitous in the literature
- Summarize $\mathcal{D}$'s interaction into a transcript τ, and let

$$p_{\mathrm{re}}(\tau) = \mathbf{Pr}\left(\mathcal{D}^{\mathcal{O}} \text{ yields } \tau\right), \qquad p_{\mathrm{id}}(\tau) = \mathbf{Pr}\left(\mathcal{D}^{\mathcal{I}} \text{ yields } \tau\right)$$

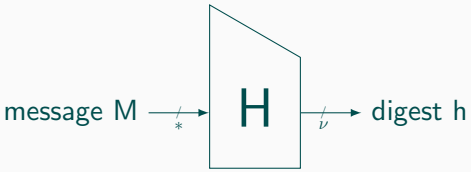
- Split attainable τ into $\mathcal{T}_{\mathrm{good}} \sqcup \mathcal{T}_{\mathrm{bad}}$. We have

$$\frac{p_{\mathrm{re}}(\tau)}{p_{\mathrm{id}}(\tau)} \geq 1 - \varepsilon \quad \forall \tau \in \mathcal{T}_{\mathrm{good}}, \qquad \mathbf{Pr}\left(\mathcal{D}^{\mathcal{I}} \text{ yields } \tau \in \mathcal{T}_{\mathrm{bad}}\right) \leq \delta$$
$$\Longrightarrow \mathbf{Adv}^{\mathrm{ind}}(\mathcal{D}) \leq \varepsilon + \delta$$

- Identical-until-bad: special case when $p_{\mathrm{re}}(\tau) = p_{\mathrm{id}}(\tau), \forall \tau \in \mathcal{T}_{\mathrm{good}}$
- H-coefficient is more general, conceptually a bit different

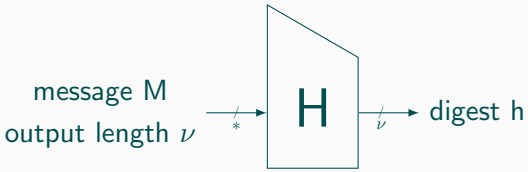
Hash Functions

Hash Functions



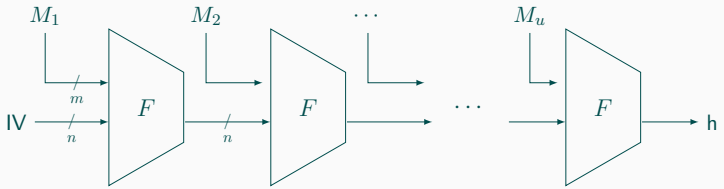
- A hash function maps an arbitrarily long message M to a fixed-size digest h

Hash Functions



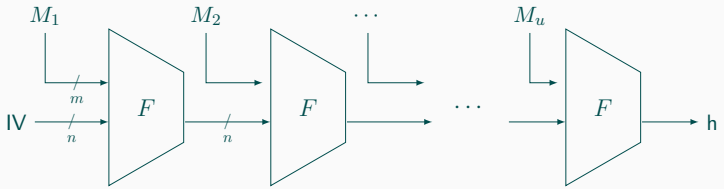
- A hash function maps an arbitrarily long message M to a fixed-size digest h
- Generalized version: eXtendable Output Function (XOF)

Merkle-Damgård (MD) Construction [Mer89, Dam89]

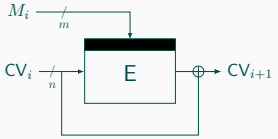


- Construction underlying the SHA-2 family
- Message is injectively padded and split into m-bit blocks, which are processed iteratively via a compression function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$

Merkle-Damgård (MD) Construction [Mer89, Dam89]

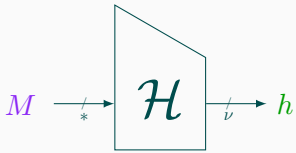


- Construction underlying the SHA-2 family
- Message is injectively padded and split into m-bit blocks, which are processed iteratively via a compression function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$
- F can be built with block ciphers, e.g., Davies-Meyer



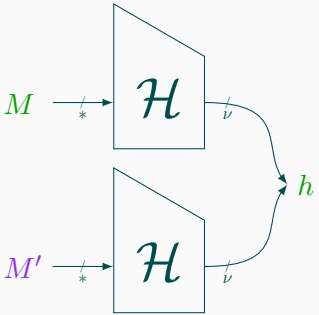
Hash Function Security: Pre/SecPre/Col (1/2)

Preimage Resistance



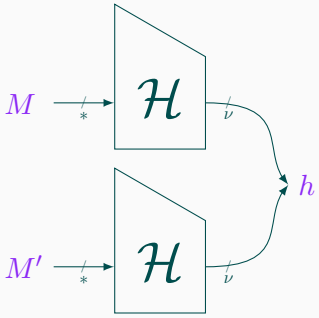
Given h, it should be hard to find M

Second Preimage Resistance



Given M, it should be hard to find $M' \neq M$ such that $\mathcal{H}(M) = \mathcal{H}(M')$

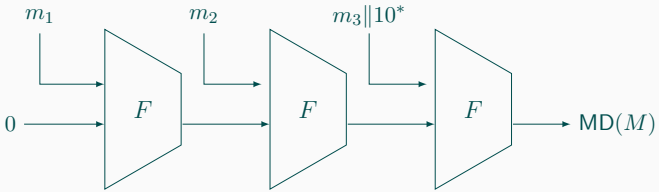
Collision Resistance



It should be hard to find $M' \neq M$ such that $\mathcal{H}(M) = \mathcal{H}(M')$

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.

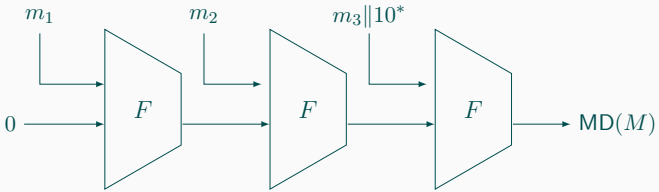


$M = m_1 \| m_2 \| m_3$

- MD^F with a random F is Pre/SecPre/Col

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.

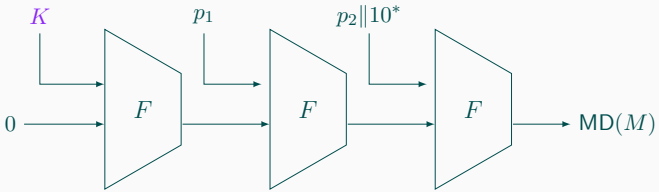


$M = m_1 \| m_2 \| m_3$

- MD^F with a random F is Pre/SecPre/Col
- Does not imply security in any use-case due to length extensions attacks

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.

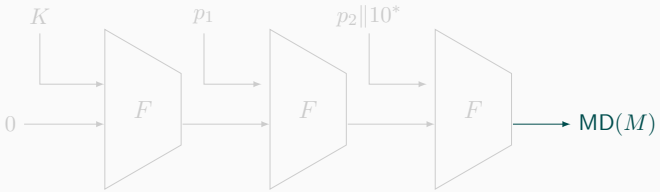


$M = K \| p_1 \| p_2$

- MD^F with a random F is Pre/SecPre/Col
- Does not imply security in any use-case due to length extensions attacks

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.

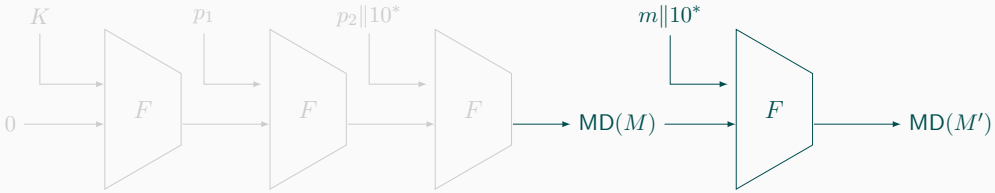


$M = K \| p_1 \| p_2$

- MD^F with a random F is Pre/SecPre/Col
- Does not imply security in any use-case due to length extensions attacks

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.

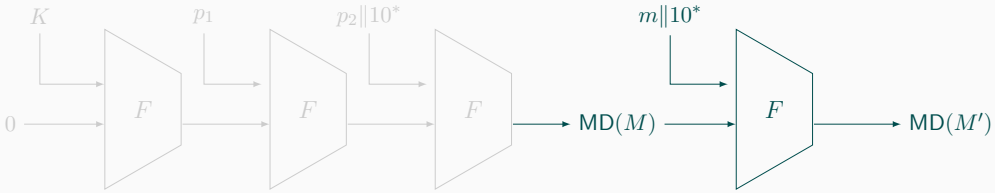


$M' = K \| p_1 \| p_2 \| 10^* \| m$

- MD^F with a random F is Pre/SecPre/Col
- Does not imply security in any use-case due to length extensions attacks

Hash Function Security: Pre/SecPre/Col (2/2)

Take MD Construction with 10^* padding: append a 1, then as many 0's as needed so the length is a multiple of m.



$M' = K \| p_1 \| p_2 \| 10^* \| m$

- MD^F with a random F is Pre/SecPre/Col
- Does not imply security in any use-case due to length extensions attacks
- We want something stronger: the construction behaves like a random oracle

Random Oracles

- A random oracle is an idealized object: on each new input, it returns a uniformly random output
- **Consistency:** querying the same input twice always returns the same output
- We will focus on random oracles with a fixed-sized output:

$$\mathcal{RO} : \{0,1\}^* \longrightarrow \{0,1\}^\nu$$

on input M, returns a uniform random string of a fixed size ν

Indifferentiability (1/4)

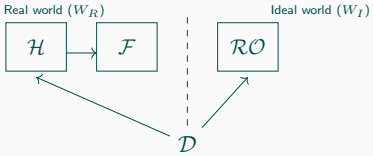
How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?

Indifferentiability (1/4)

How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?

Indifferentiability (1/4)

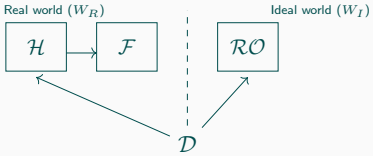
How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?



- **Attempt:** W_R gives access to $\mathcal{H}^{\mathcal{F}}$, W_I gives access to $\mathcal{RO}$

Indifferentiability (1/4)

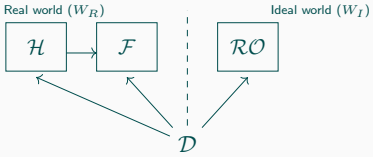
How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?



- **Attempt:** W_R gives access to $\mathcal{H}^{\mathcal{F}}$, W_I gives access to $\mathcal{RO}$
- **Problem:** in the real world, $\mathcal{D}$ could also query $\mathcal{F}$ directly

Indifferentiability (1/4)

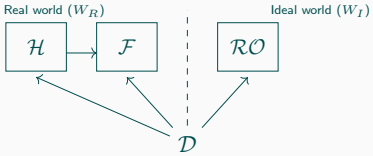
How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?



- **Attempt:** W_R gives access to $\mathcal{H}^{\mathcal{F}}$, W_I gives access to $\mathcal{RO}$
- **Problem:** in the real world, $\mathcal{D}$ could also query $\mathcal{F}$ directly
- So $\mathcal{D}$ needs access to $\mathcal{F}$ in W_R

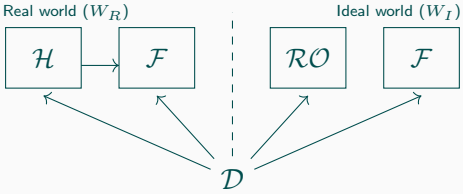
Indifferentiability (1/4)

How to prove that a hash function construction $\mathcal{H}$ based on a random primitive $\mathcal{F}$ behaves like a random oracle $\mathcal{RO}$?



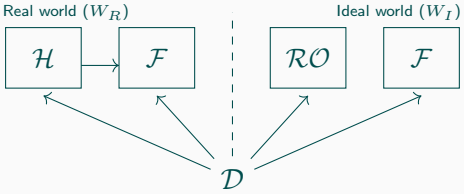
- **Attempt:** W_R gives access to $\mathcal{H}^{\mathcal{F}}$, W_I gives access to $\mathcal{RO}$
- **Problem:** in the real world, $\mathcal{D}$ could also query $\mathcal{F}$ directly
- So $\mathcal{D}$ needs access to $\mathcal{F}$ in W_R
- Which primitive should W_I implement?

Indifferentiability (2/4)



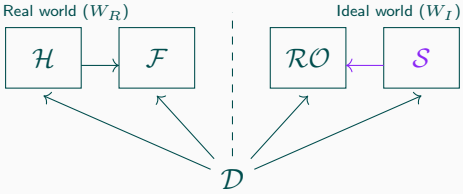
- **Attempt:** W_I's primitive oracle gives access to a random $\mathcal{F}$

Indifferentiability (2/4)



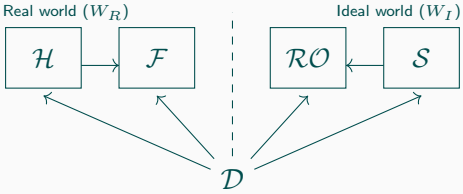
- **Attempt:** W_I's primitive oracle gives access to a random $\mathcal{F}$
- **New problem:** In W_I, we have two objects that are unrelated to each other, while there is a relation between primitive and construction queries in W_R

Indifferentiability (2/4)



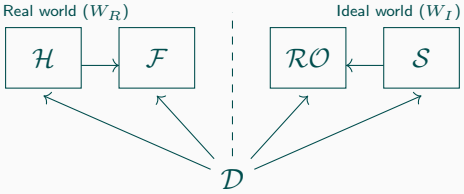
- **Attempt:** W_I's primitive oracle gives access to a random $\mathcal{F}$
- **New problem:** In W_I, we have two objects that are unrelated to each other, while there is a relation between primitive and construction queries in W_R
- **Solution:** [MRH04, CDMP05] replace primitive oracle with a *simulator* $\mathcal{S}$ that plays the role of $\mathcal{F}$ in the ideal world, while staying consistent with $\mathcal{RO}$

Indifferentiability (3/4)



- $\mathcal{H}$ is indifferentiable from $\mathcal{RO}$ for some simulator $\mathcal{S}$ whenever any $\mathcal{D}$ can distinguish the two worlds only with a negligible probability
- Advantage is denoted as $\mathbf{Adv}^{\text{indiff}}_{\mathcal{H},\mathcal{S}}(\mathcal{A})$

Indifferentiability (4/4)



- For any single-stage game,[3] indifferentiability allows to replace $(\mathcal{H}^{\mathcal{F}}, \mathcal{F})$ with $(\mathcal{RO}, \mathcal{S}[\mathcal{RO}])$

$\Longrightarrow$ Attacking a cryptosystem $\mathcal{C}$ based on $\mathcal{H}^{\mathcal{F}}$ reduces to attacking $\mathcal{C}$ based on $\mathcal{RO}$, at the cost of the indifferentiability advantage

- In particular, for $\mathsf{atk} \in \{\mathsf{pre}, \mathsf{secpre}, \mathsf{col}\}$, we have [AMP10]

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

[3] See [RSS11] for examples where composition fails

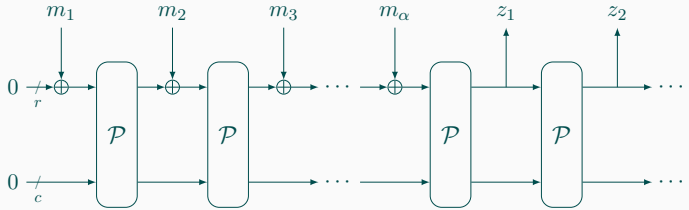
The Sponge Construction and Its Indifferentiability Proof

The Sponge and Cryptographic Competitions

Also: Arithmetization-Oriented Sponges, e.g., [GKR+21, BBC+23, GKL+22]

- 2007 – Introduction of sponge [BDPV07]
- 2008 – SHA-3 competition begins
  - Call for hashing
- 2011 – Introduction of the duplex with SpongeWrap [BDPV11a]
- 2012 – SHA-3 competition ends (FIPS PUB 202)
  - Keccak selected as winner (SHA3-{224,256,384,512}, SHAKE{128,256})
- 2014 – CAESAR competition begins
  - Call for AEAD; 57 submissions, of which $\approx$ 10 sponge/duplex-based
- 2019 – CAESAR competition ended
  - Ascon won in the category **lightweight**
- 2019 – NIST lightweight crypto competition begins
  - Call for AEAD (optional hashing)
  - 57 submissions, of which $\approx$ 22 sponge/duplex-based
- 2023 – NIST Lightweight crypto competition ends (NIST SP 800-232)
  - Ascon selected as overall winner

The Sponge Construction



- Permutation of size $b = r + c$ bits:
  - rate r: outer part of the state (efficiency parameter)
  - capacity c: inner part of the state (security parameter)
- The message is padded with a sponge-compliant padding (e.g., 10^*)

Generic Security of the Sponge [BDPV08]

- There exists $\mathcal{S}$ (using at most q $\mathcal{RO}$-queries) s.t., for any $\mathcal{D}$ with query budget q:

$$\mathbf{Adv}^{\text{indiff}}_{\text{Sponge},\mathcal{S}}(\mathcal{D}) \leq \frac{q(q+2)}{2^c}$$

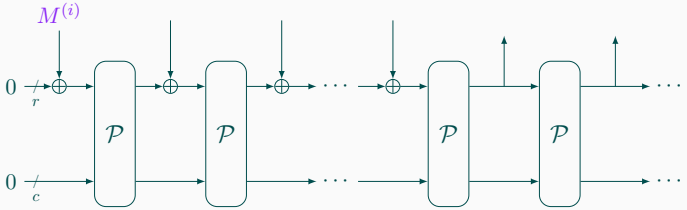
- Implies security of sponge truncated to ν bits against classical attacks:

$$\mathbf{Adv}^{\text{Col}}_{\text{Sponge}}(\mathcal{A}) \leq \frac{q(q+2)}{2^c} + \frac{q(q-1)}{2^{\nu+1}} \qquad \mathbf{Adv}^{\text{SecPre}}_{\text{Sponge}}(\mathcal{A}) \leq \frac{q(q+2)}{2^c} + \frac{q}{2^\nu}$$

$$\mathbf{Adv}^{\text{Pre}}_{\text{Sponge}}(\mathcal{A}) \leq \frac{q(q+2)}{2^c} + \frac{q}{2^\nu}$$

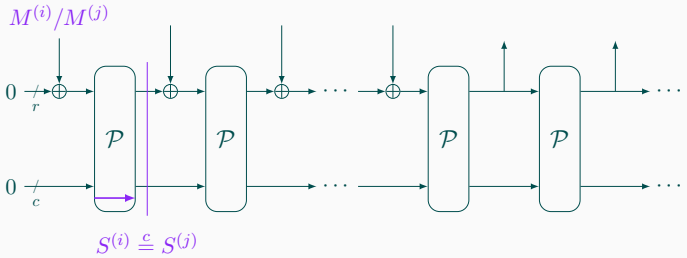
- Attacks already described in [BDPV07]
- Col, SecPre bounds tight; Pre bound needed a dedicated proof [LM22]

Collision Attack [BDPV07]



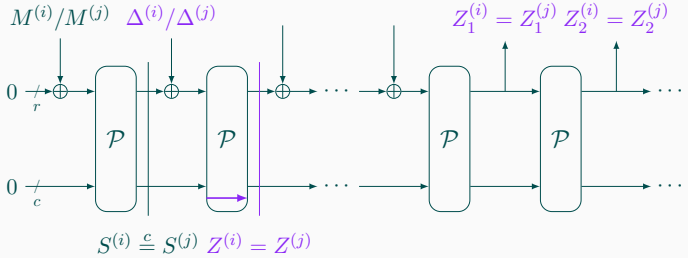
- Absorb $2^{c/2}$ different message blocks

Collision Attack [BDPV07]



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

Collision Attack [BDPV07]

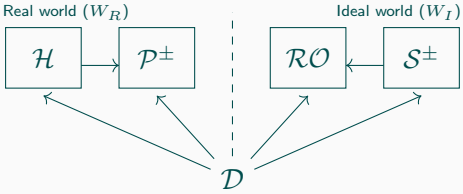


- Absorb $2^{c/2}$ different message blocks
- With high probability, there exists $S^{(i)} \neq S^{(j)}$ that collide on their inner part
- Compensate the difference in the outer part with $\Delta^{(i)} = \lceil S^{(i)} \rceil_r$, $\Delta^{(j)} = \lceil S^{(j)} \rceil_r$
- Triggers a full-state collision $\implies$ collision for any output length

The Sponge Construction and Its Indifferentiability Proof

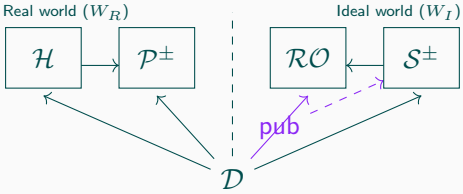
Indifferentiability Proof

Reminder about Indifferentiability



- $\mathcal{P}$ is a random permutation with bidirectional access, $\mathcal{H}$ denotes the sponge construction
- Goal: build a simulator $\mathcal{S}$ such that $(\mathcal{H}^{\mathcal{P}}, \mathcal{P}^{\pm}) \approx (\mathcal{RO}, (\mathcal{S}^{\mathcal{RO}})^{\pm})$ for any $\mathcal{D}$ with reasonable query complexity
- q denotes $\mathcal{D}$'s total query budget if we were dealing with the real world
- Here, $\mathcal{S}$ does not know the queries that $\mathcal{D}$ made to $\mathcal{RO}$

Public Indifferentiability [YMO09, DRS09, MPS12] (1/2)



- A notion weaker than plain indifferentiability: here, $\mathcal{S}$ additionally sees the adversary's construction queries
- Consequence: we can assume that each construction query directly triggers a call to $\mathcal{S}$ with the matching input

Public Indifferentiability [YMO09, DRS09, MPS12] (2/2)

- Public indifferentiability is weaker: (plain) Merkle–Damgård is publicly indifferentiable but not indifferentiable [DRS09]
- Yet it is remains meaningful and powerful notion, and allows composition whenever inputs to $\mathcal{H}$ are not meant to be secret
- We have

$$\text{Indiff} \implies \text{Public Indiff} \underset{[\text{MPS12}]}{\implies} \text{Correlation Intractability} \underset{[\text{CGH98}]}{\implies} \text{Col/Pre/SecPre}$$

- **In our case**: used as an intermediate step to prove full indifferentiability

Simplifying assumption

- One squeezed block per message
- Inputs are assumed pre-padded

Notation

- $\mathtt{C}$: construction oracle $\mathtt{P}$: primitive oracle

Roadmap

(1) Build a simulator

(2) Graph representation and analysis $\Longrightarrow$ consistency of the answers

(3) Split the analysis with an intermediate world $\Longrightarrow$ split quality of randomness of the simulator and consistency

(4) Conclude and get public indifferentiability

(5) Upgrade the result to indifferentiability

Simulator Intuition (1/2)

Let $m \in \{0,1\}^r$. Assume the adversary makes the queries

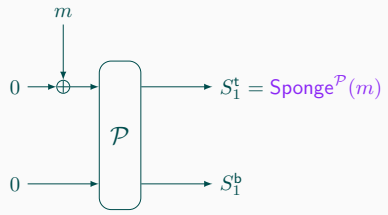
$$\mathtt{P}(m \| 0^c) \qquad \text{and} \qquad \mathtt{C}(m)$$

Simulator Intuition (1/2)

Let $m \in \{0,1\}^r$. Assume the adversary makes the queries

$$\mathtt{P}(m \| 0^c) \qquad \text{and} \qquad \mathtt{C}(m)$$

Real World: C uses P



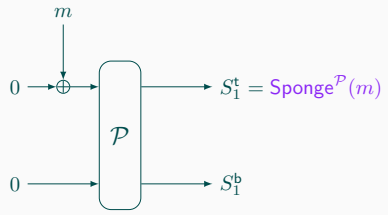
$$\lceil \mathcal{P}(m \| 0^c) \rceil_r = \mathsf{Sponge}^{\mathcal{P}}(m)$$

Simulator Intuition (1/2)

Let $m \in \{0,1\}^r$. Assume the adversary makes the queries

$$\mathtt{P}(m\|0^c) \qquad \text{and} \qquad \mathtt{C}(m)$$

Real World: C uses P



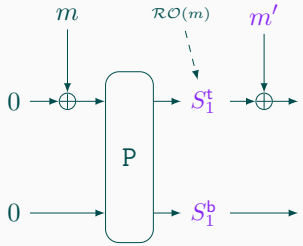
$$\lceil \mathcal{P}(m\|0^c) \rceil_r = \mathsf{Sponge}^{\mathcal{P}}(m)$$

Ideal World:

- C is a random oracle
- So $\mathcal{S}$ takes over the consistency: outer part of its answer is fixed to $\mathcal{RO}(m)$
- Inner part: random (will change later!)

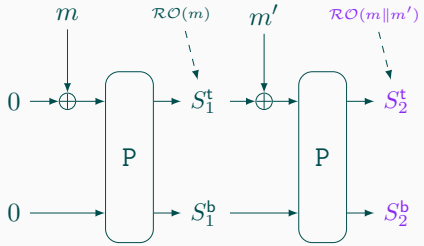
$\Longrightarrow \quad \mathcal{S}(m\|0^c) = \mathcal{RO}(m)\|\mathtt{rand}$

Simulator Intuition (2/2)



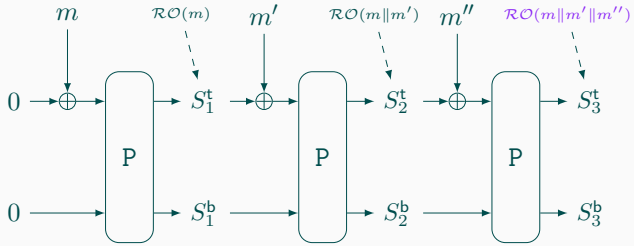
- Assume next query is of the form $\mathtt{P}(S_1^t \| S_1^b \oplus (m' \| 0^c))$

Simulator Intuition (2/2)



- Assume next query is of the form $\mathtt{P}(S_1^t \| S_1^b \oplus (m' \| 0^c))$
- Simulator needs to reply with a state whose outer part is consistent with $\mathcal{RO}(m\|m')$

Simulator Intuition (2/2)



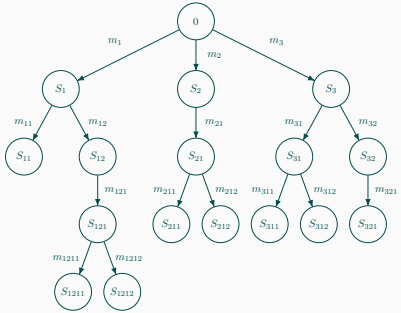
- Assume next query is of the form $\mathtt{P}(S_1^t \| S_1^b \oplus (m' \| 0^c))$
- Simulator needs to reply with a state whose outer part is consistent with $\mathcal{RO}(m\|m')$
- This goes on for each element on the chain

Simulator Intuition (2/2)

- Assume next query is of the form $\mathtt{P}(S_1^t \| S_1^b \oplus (m' \| 0^c))$
- Simulator needs to reply with a state whose outer part is consistent with $\mathcal{RO}(m \| m')$
- This goes on for each element on the chain
- This induces a graph construction

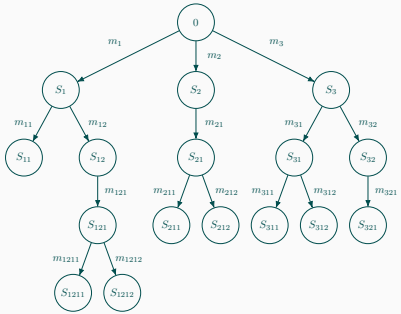
Graph Construction and Consistency of the Answers (1/2)

- Nodes represent states, edges absorption of a message block
- A path from 0 is a sequence of message blocks

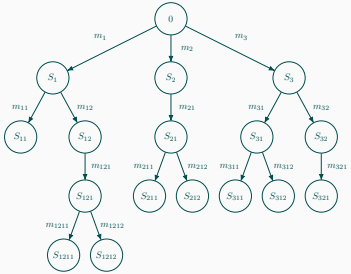


Graph Construction and Consistency of the Answers (1/2)

- Nodes represent states, edges absorption of a message block
- A path from 0 is a sequence of message blocks
- $\mathcal{S}$ acts as follows:
  - ◇ On a forward query, $\mathcal{S}$ checks if inner part appears in a node reachable from 0;
    if so, it reconstructs the associated message, and answers consistently with $\mathcal{RO}$;
    else, answer at random
  - ◇ On a inverse query, random answer

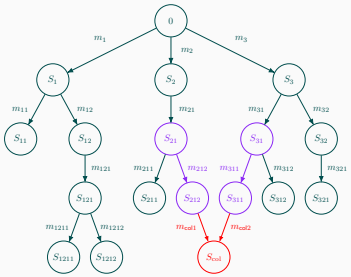


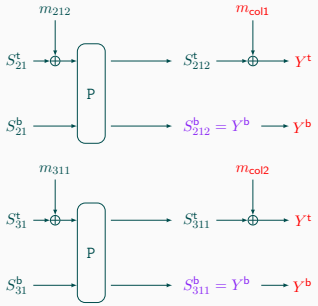
Graph Construction and Consistency of the Answers (2/2)

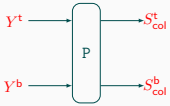


What can go wrong?

Graph Construction and Consistency of the Answers (2/2)



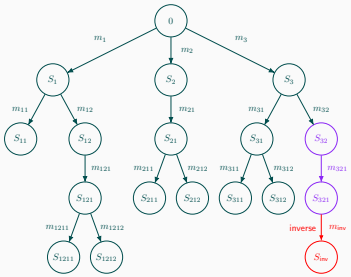


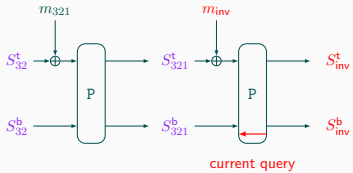


What can go wrong?

- **Collision** between two rooted nodes

Graph Construction and Consistency of the Answers (2/2)

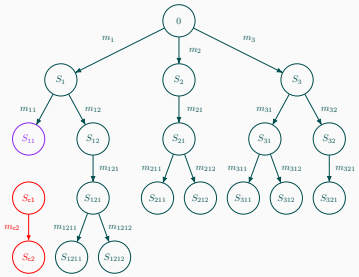




What can go wrong?

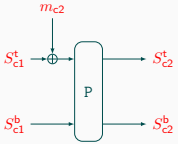
- **Collision** between two rooted nodes
- **A posteriori connection**

Graph Construction and Consistency of the Answers (2/2)



S_{11}^{t}

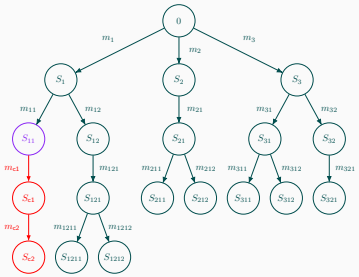
S_{11}^{b}

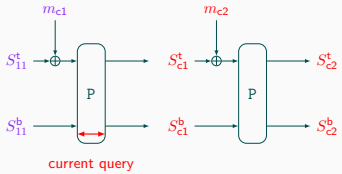


What can go wrong?

- **Collision** between two rooted nodes
- **A posteriori connection**

Graph Construction and Consistency of the Answers (2/2)

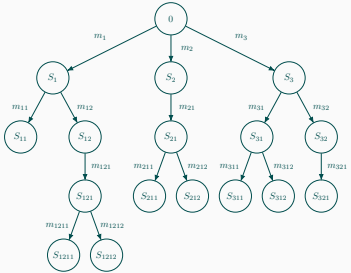




What can go wrong?

- **Collision** between two rooted nodes
- **A posteriori connection**

Graph Construction and Consistency of the Answers (2/2)



What can go wrong?

- **Collision** between two rooted nodes
- **A posteriori connection**

✓ Two possibilities:

- Turn this into bad event, analyze proba of bad
- Or like [BDPV08]: ensure that the inner parts are fresh $\implies$ we will do that

The Simulator from [BDPV08]

Algorithm 3 The permutation simulator $\mathcal{P}[\mathcal{RO}]$

Interface $\mathcal{F}^{1}$, taking node s as input
if node s has no outgoing edge **then**
  if node s is rooted AND $R \cup O \neq C$ (no saturation) **then**
    Construct path to t: find path to s, append s_{a} and call the result p
    Write p as $p = p'0^{r'j}$ where p' does not end with 0^{r}
    if p' can be unpadded into x **then**
      Assign to t_{a} the value z_j with $z = \mathcal{RO}(x)$
    else
      Choose t_{a} randomly and uniformly
    end if
    Choose t_{c} randomly and uniformly from $C \setminus (R \cup O)$ and such that $(t_{\mathrm{a}}, t_{\mathrm{c}})$ has no incoming edge yet
    Let $t = (t_{\mathrm{a}}, t_{\mathrm{c}})$
  else
    Choose t randomly and uniformly from all nodes that have no incoming edge yet
  end if
  Add an edge from s to t
end if
return the node t at the end of the outgoing edge from s

Interface $\mathcal{F}^{-1}$, taking node s as input
if node s has no incoming edge **then**
  Choose t_{a} randomly and uniformly
  Choose t_{c} randomly and uniformly from $C \setminus R$ and such that $(t_{\mathrm{a}}, t_{\mathrm{c}})$ has no outgoing edge yet
  Let $t = (t_{\mathrm{a}}, t_{\mathrm{c}})$
  Add an edge from t to s
end if
return the node t at the beginning of the incoming edge into s

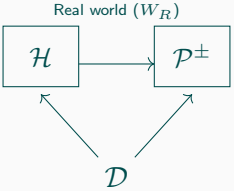
The Simulator is Consistent

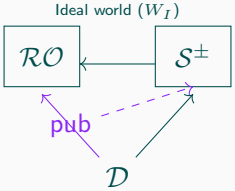
Lemma (consistency of the answers)

As long as $q < 2^c$, whenever $\mathsf{Sponge}^{\mathcal{S}^{\mathcal{RO}}}(M)$ is computable from the simulator query history, we have

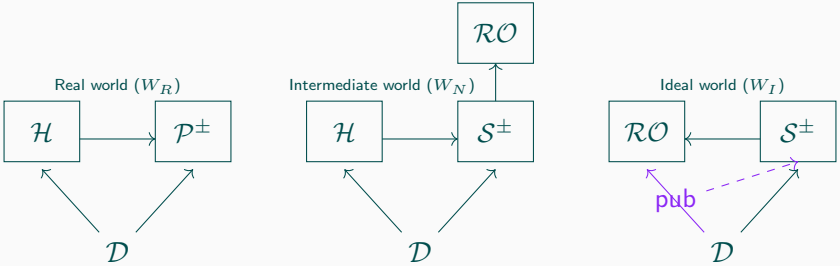
$$\mathsf{Sponge}^{\mathcal{S}^{\mathcal{RO}}}(M) = \mathcal{RO}(M)$$

World Decomposition

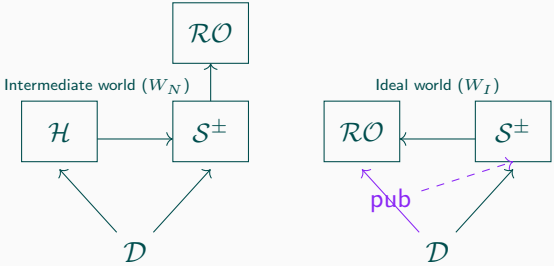




World Decomposition

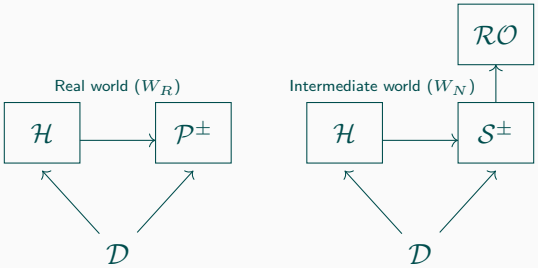


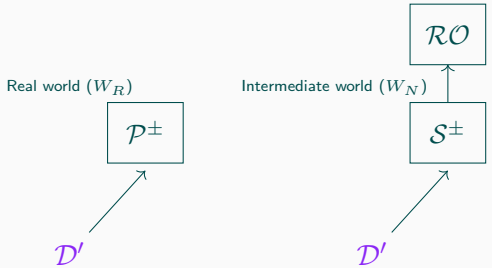
- Introduce an intermediate world W_N where:
  - Primitive queries implemented by $\mathcal{S}$ based on $\mathcal{RO}$ hidden from $\mathcal{D}$
  - Construction queries: Sponge based on $\mathcal{S}$
- Two distances to analyze



- $\mathcal{S}$ is aware of all construction queries made by $\mathcal{D}$
- Along with the consistency lemma, querying directly $\mathcal{H}$ or $\mathcal{RO}$ does not make a difference, the answers will be the same

  $\implies$ As long as $q < 2^c$, $\Delta_{\mathcal{D}}(W_N; W_I) = 0$


Real world (W_R)
$\mathcal{H}$
$\mathcal{P}^{\pm}$
$\mathcal{D}$
Intermediate world (W_N)
$\mathcal{RO}$
$\mathcal{H}$
$\mathcal{S}^{\pm}$
$\mathcal{D}$



- We can drop the construction oracle by building a new adversary $\mathcal{D}'$ from $\mathcal{D}$
- The distinguishing game reduces to analyzing the quality of randomness of $\mathcal{S}$
- An analysis gives $\Delta_{\mathcal{D}}(W_N; W_R) \leq \frac{q(q+1)}{2^c}$

Conclusion: Public Indifferentiability

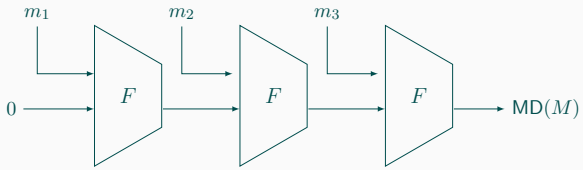
- We obtain

$$\mathbf{Adv}^{\text{pubiff}}_{\text{Sponge},\mathcal{S}}(\mathcal{D}) \leq \frac{q(q+1)}{2^c}$$

- Simulator makes at most q queries to $\mathcal{RO}$
- The proof extends to arbitrary-length squeezing, provided the padding rule is sponge-compliant
- Last step: generalize to full indifferentiability

Stretching to Full Indifferentiability (1/3)

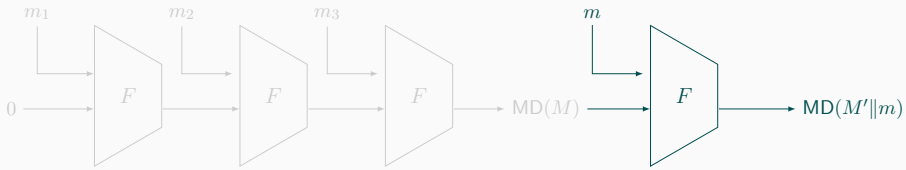
Going back to (plain) Merkle–Damgård:



- Can be shown publicly indiff with advantage $\approx \frac{q^2}{2^n}$ using a similar technique

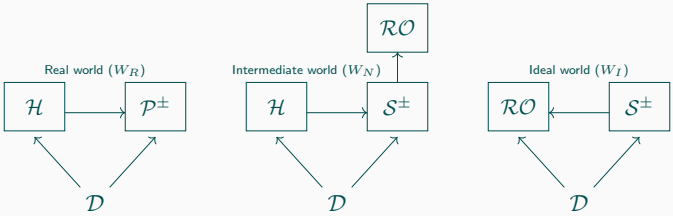
Stretching to Full Indifferentiability (1/3)

Going back to (plain) Merkle–Damgård:



- Can be shown publicly indiff with advantage $\approx \frac{q^2}{2^n}$ using a similar technique
- However, (plain) MD is broken by length extension attacks, does not behave like $\mathcal{RO} \implies$ not indifferentiable
- Resistance against length extension attacks is the missing piece of our proof

Stretching to Full Indifferentiability (2/3)



- In our setting, the difference occurs in the distance between W_N and W_I:
  - In W_N the simulator has more information due to the queries made to $\mathcal{H}$
  - Need to show that this extra knowledge does not put $\mathcal{S}$ from W_I at disadvantage

Stretching to Full Indifferentiability (3/3)

- Assume $\mathcal{D}$ makes a construction query

Stretching to Full Indifferentiability (3/3)

- Assume $\mathcal{D}$ makes a construction query $\implies$ In W_N, $\mathcal{S}$'s graph construction is expanded

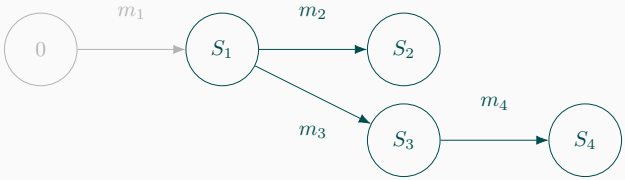
Stretching to Full Indifferentiability (3/3)

- Assume $\mathcal{D}$ makes a construction query $\implies$ In W_N, $\mathcal{S}$'s graph construction is expanded
- Assume $\mathcal{D}$ then makes the query $\mathtt{P}(S_1 \oplus m_2 \| 0^c)$

Stretching to Full Indifferentiability (3/3)

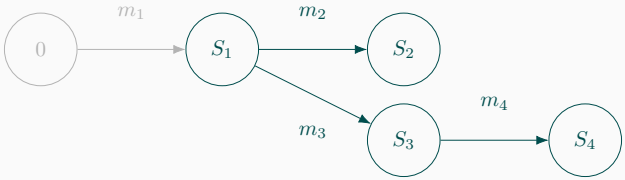
- Assume $\mathcal{D}$ makes a construction query $\implies$ In W_N, $\mathcal{S}$'s graph construction is expanded
- Assume $\mathcal{D}$ then makes the query $\mathtt{P}(S_1 \oplus m_2 \| 0^c) \implies$ In W_N, the answer will be consistent

Stretching to Full Indifferentiability (3/3)



- Assume $\mathcal{D}$ makes a construction query $\implies$ In W_N, $\mathcal{S}$'s graph construction is expanded
- Assume $\mathcal{D}$ then makes the query $\mathtt{P}(S_1 \oplus m_2 \| 0^c) \implies$ In W_N, the answer will be consistent, and so is any P-query extending from S_1

Stretching to Full Indifferentiability (3/3)



- Assume $\mathcal{D}$ makes a construction query $\implies$ In W_N, $\mathcal{S}$'s graph construction is expanded
- Assume $\mathcal{D}$ then makes the query $\mathtt{P}(S_1 \oplus m_2 \| 0^c) \implies$ In W_N, the answer will be consistent, and so is any P-query extending from S_1
- But such a situation does not happen in $W_I \implies$ exploitable to distinguish
- Add a bad event in W_N of the form "$\mathcal{D}$ queries a rooted node that was not built from left to right via permutation queries" $\implies$ extra term of the form $\frac{q}{2^c}$

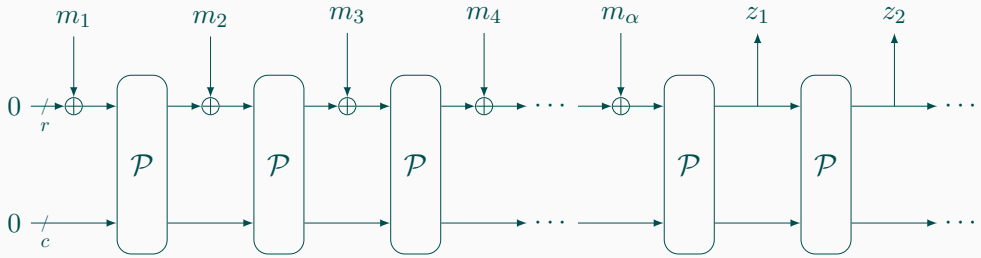
Indifferentiability: Conclusion

The sponge is indifferentiable from a random oracle:

- Designed a simulator $\mathcal{S}$, track its answers via a graph, and argued $\mathcal{RO}$-consistency
- Built an intermediate world W_N, and split the analysis via the triangle inequality
- This gave public indifferentiability
- One extra bad event upgrades the bound to full indifferentiability

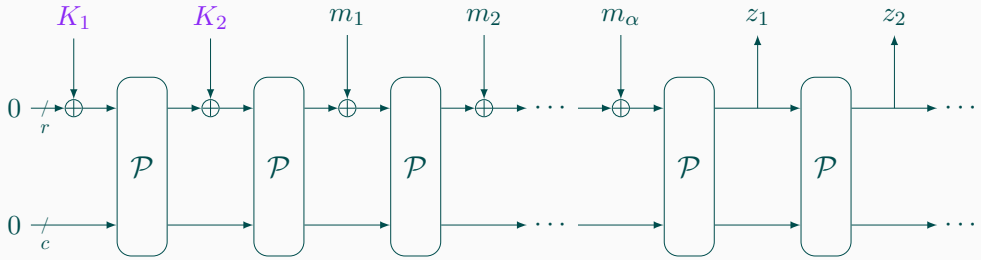
Keyed Applications of Sponges

Sponge-Based PRF Constructions



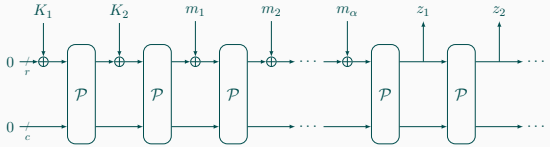
- The sponge can be tweaked

Sponge-Based PRF Constructions



- The sponge can be tweaked to build a pseudorandom function
- Blackbox invokation of sponge with input $K \| M$
- The construction is named Outer-Keyed Sponge [BDPV11b]

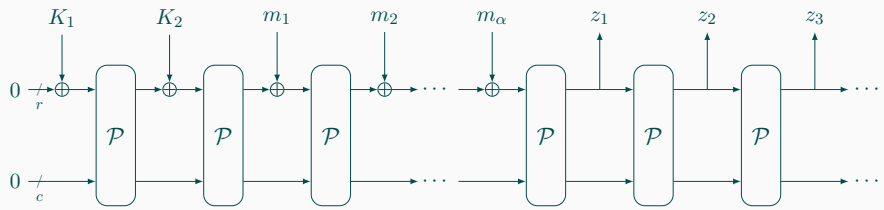
Outer Keyed Sponge: Security



- PRF security of OKS can be deduced from indifferentiability ...
- ... but indifferentiability does not distinguish between offline and online complexities
- A dedicated analysis yields a tighter bound [GPT15, ADMV15, NY16, Men18]. Ignoring constant/log factors:

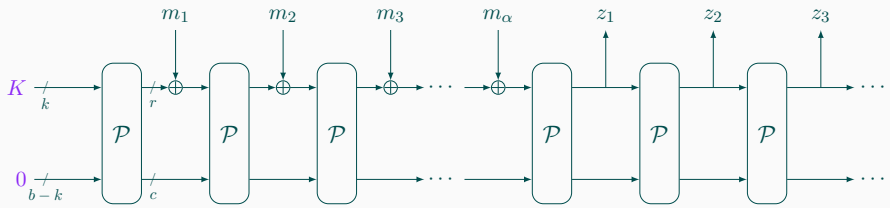
$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{OKS}}(\mathcal{D}) = \tilde{\mathcal{O}}\left(\frac{q_{\mathsf{off}}}{2^k} + \frac{q_{\mathsf{off}} q_{\mathsf{on}}}{2^c}\right)$$

Sponge-Based PRF Constructions: Optimization



A long line of research showed:

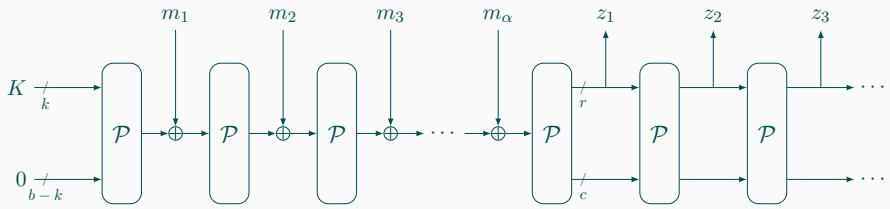
Sponge-Based PRF Constructions: Optimization



A long line of research showed:

- State may be initialized with key, in various ways [BDPV12, DM24], ...

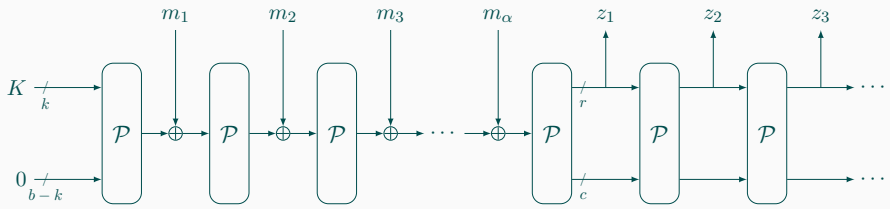
Sponge-Based PRF Constructions: Optimization



A long line of research showed:

- State may be initialized with key, in various ways [BDPV12, DM24], ...
- May absorb over the entire state [BDPV11c, BDPV12, CDH+12, ADMV15], ...

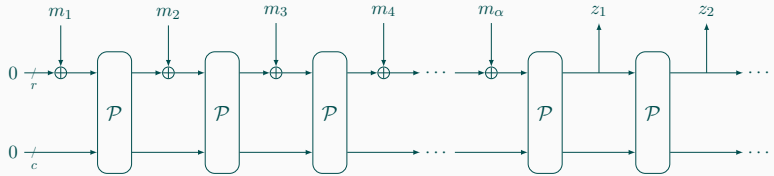
Sponge-Based PRF Constructions: Optimization



A long line of research showed:

- State may be initialized with key, in various ways [BDPV12, DM24], ...
- May absorb over the entire state [BDPV11c, BDPV12, CDH$^+$12, ADMV15], ...
- ... without decreasing generic security. Also, security bounds that capture fine-grained adversarial power were conducted

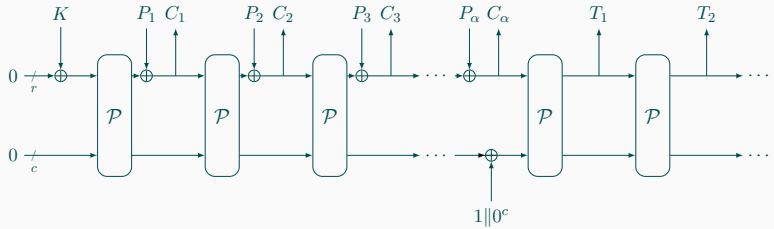
Sponge-Based AEAD Constructions



- The sponge can be tweaked

[4] Simplified for the figure, see [BDPV11a] for the exact construction

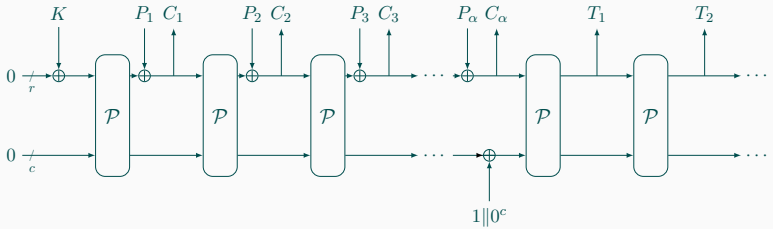
Sponge-Based AEAD Constructions



- The sponge can be tweaked for authenticated encryption purposes

[4] Simplified for the figure, see [BDPV11a] for the exact construction

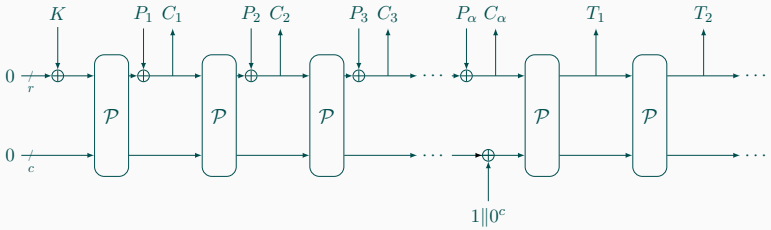
Sponge-Based AEAD Constructions



- The sponge can be tweaked for authenticated encryption purposes
- The first construction to achieve that is named SpongeWrap [BDPV11a][4]

[4] Simplified for the figure, see [BDPV11a] for the exact construction

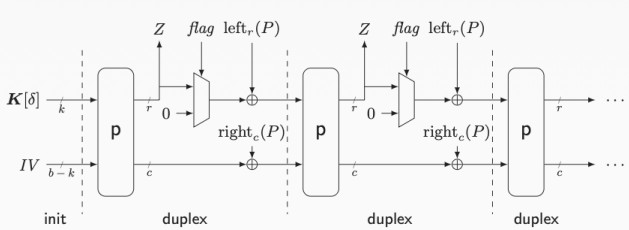
Sponge-Based AEAD Constructions



- The sponge can be tweaked for authenticated encryption purposes
- The first construction to achieve that is named SpongeWrap [BDPV11a][4]
- Build on top of the duplex [BDPV11a], a sort of online version of the sponge

[4] Simplified for the figure, see [BDPV11a] for the exact construction

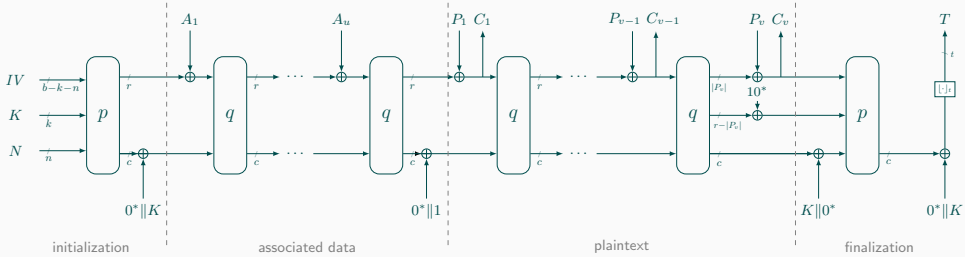
Keyed Duplex



(picture taken from Bart Mennink's talk Understanding the Duplex and Its Security) [Men23]

- Very general object that captures a lot of use-cases for keyed applications[5]
- Bounds parametrized by various use-cases, complex to interpret

[5] Unkeyed versions of duplex were also treated, see [DFG23]



Variant of SpongeWrap

- Outer permutation p and inner permutation q, both on b bits
- Additional key blindings around "outer" permutations $\implies$ better authenticity and strong leakage resilience
- See SoK: Security of the Ascon Modes [LM25] for a broad picture on Ascon's generic security

Conclusion

Conclusion

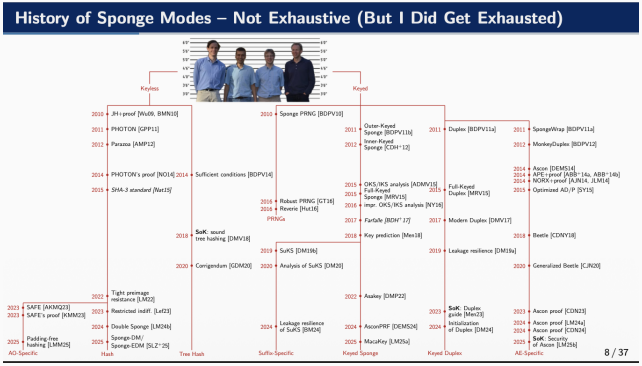
- The sponge is a versatile and cool paradigm
- Its indifferentiability result is central, but sometimes we want dedicated analysis
- Proofs in random permutation model
  - Sometimes possible to have standard model proofs (e.g., reduce to EM for keyed sponges)
  - Weakening ideal permutation model is an interesting open problem
- Provable security/mode design of permutation based crypto is still active, e.g.,
  - Permutation-Based Hashing with security "Beyond the Birthday Bound", e.g., [LM24, LSY+26, LLHL26]
  - Security of sponges against a quantum computer, e.g., [ACMT25, Hos25, CHL+25]
  - And many more $\cdots$
- Machine-checked proofs (e.g, for sponge [ABB+19])

FSE 2026 invited talk

Evolution of Permutation-Based Modes and Their Security

Bart Mennink

(slides: on his website, recording will follow)



Online GAPS

- Recurring online seminars on provable security for symmetric-key cryptography since February 2026
- Website: https://gapsworkshop.github.io/online/ (contains links to slides and recording of most past presentations)
- You may also join the google group: https://groups.google.com/g/onlinegaps

Website:

Online GAPS

- Recurring online seminars on provable security for symmetric-key cryptography since February 2026
- Website: https://gapsworkshop.github.io/online/ (contains links to slides and recording of most past presentations)
- You may also join the google group: https://groups.google.com/g/onlinegaps

Website:

Thank you for your attention!

[ABB+19] José Bacelar Almeida, Cécile Baritel-Ruet, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Alley Stoughton, and Pierre-Yves Strub.
Machine-Checked Proofs for Cryptographic Standards: Indifferentiability of Sponge and Secure High-Assurance Implementations of SHA-3.
In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 1607–1622. ACM, 2019.

[ACMT25] Gorjan Alagic, Joseph Carolan, Christian Majenz, and Saliha Tokat.
The Sponge Is Quantum Indifferentiable.
In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 2591–2630. IEEE, 2025.

[ADMV15] 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.

[AMP10] 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.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems.
New Design Techniques for Efficient Arithmetization-Oriented Hash Functions: ttAnemoi Permutations and ttJive Compression Mode.
In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539. Springer, 2023.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Sponge Functions.
Ecrypt Hash Workshop 2007, May 2007.

[BDPV08] 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.

[BDPV11a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications.
In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[BDPV11b] 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.

[BDPV11c] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
On the Security of the Keyed Sponge Construction.
Symmetric Key Encryption Workshop, February 2011.

[BDPV12] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Permutation-based encryption, authentication and authenticated encryption.
Directions in Authenticated Ciphers, July 2012.

[BR06] Mihir Bellare and Phillip Rogaway.
The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs.
In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[CDH+12] 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.

[CDMP05] 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.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi.
The Random Oracle Methodology, Revisited (Preliminary Version).
In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 209–218. ACM, 1998.

[CHL+25] Alexandru Cojocaru, Minki Hhan, Qipeng Liu, Takashi Yamakawa, and Aaram Yun.
Quantum Lifting for Invertible Permutations and Ideal Cipher.
In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings,*

Part II, volume 16001 of *Lecture Notes in Computer Science*, pages 481–512. Springer, 2025.

[CS14] Shan Chen and John P. Steinberger.
Tight Security Bounds for Key-Alternating Ciphers.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, 2014.

[Dam89] Ivan Damgård.
A Design Principle for Hash Functions.
In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[DFG23] Jean Paul Degabriele, Marc Fischlin, and Jérôme Govinden.
The Indifferentiability of the Duplex and Its Practical 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 237–269. Springer, 2023.

[DM24] 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.

[DRS09] Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. **Salvaging Merkle-Damgård for Practical Applications.** In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2009.

[EM91] Shimon Even and Yishay Mansour.
A Construction of a Cipher From a Single Pseudorandom Permutation.
In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan, November 11-14, 1991, Proceedings*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 1991.

[EM97] Shimon Even and Yishay Mansour.
A Construction of a Cipher from a Single Pseudorandom Permutation.
J. Cryptol., 10(3):151–162, 1997.

[GKL+22] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch.
Reinforced Concrete: A Fast Hash Function for Verifiable Computation.
In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1323–1335. ACM, 2022.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger.
Poseidon: A New Hash Function for Zero-Knowledge Proof Systems.
In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 519–535. USENIX Association, 2021.

[GPT15] 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.

[Hos25] Akinori Hosoyamada.
Post-Quantum Security of Keyed Sponge-Based Constructions Through a Modular Approach.
In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part I*, volume 16245 of *Lecture Notes in Computer Science*, pages 411–445. Springer, 2025.

[LLHL26] Hongli Li, Changlun Li, Honggang Hu, and Fengmei Liu.
Indifferentiability of the Sponge Hash Family: New Attacks and the Optimal Construction.
IACR Trans. Symmetric Cryptol., 2026(1):31–75, 2026.

[LM22] 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.

[LM24] Charlotte Lefevre and Bart Mennink.
Permutation-Based Hashing Beyond the Birthday Bound.
IACR Trans. Symmetric Cryptol., 2024(1):71–113, 2024.

[LM25] Charlotte Lefevre and Bart Mennink.
SoK: Security of the Ascon Modes.
IACR Trans. Symmetric Cryptol., 2025(1):138–210, 2025.

[LSY+26] Ziyang Luo, Yaobin Shen, Hailun Yan, Lei Wang, and Dawu Gu.
Alternating Sponge: A Low-Memory Hash Function with Beyond-Birthday-Bound Security.
Cryptology ePrint Archive, Paper 2026/707, 2026.

[Men18] Bart Mennink.
Key Prediction Security of Keyed Sponges.
IACR Trans. Symmetric Cryptol., 2018(4):128–149, 2018.

[Men23] Bart Mennink.
Understanding the Duplex and Its Security.
IACR Trans. Symmetric Cryptol., 2023(2):1–46, 2023.

[Mer89] Ralph C. Merkle.
One Way Hash Functions and DES.
In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin.
On the public indifferentiability and correlation intractability of the 6-round feistel construction.
In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 285–302. Springer, 2012.

[MRH04] 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.

[NY16] 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.

[Pat08] Jacques Patarin.
The "Coefficients H" Technique.
In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton.
Careful with Composition: Limitations of the Indifferentiability Framework.
In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011.

[YMO09] Kazuki Yoneyama, Satoshi Miyagawa, and Kazuo Ohta.
Leaky Random Oracle.
IEICE Trans. Fundam. Electron. Commun. Comput. Sci., 92-A(8):1795–1807, 2009.