How to Build a Short-Input Random Oracle *from* Public Random Permutations

Ritam Bhaumik, Nilanjan Datta, Avijit Dutta, Ashwin Jha, Sougata Mandal, Bart Mennink, Hrithik Nandi, Yaobin Shen

Eurocrypt 2026
Roma, Lazio, Italia

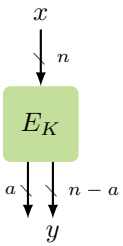
May 12, 2026

The Birthday Bound Problem

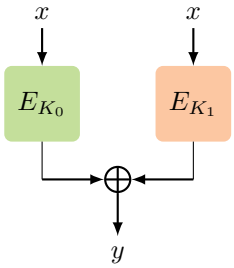
- Block ciphers: the most popular building blocks in symmetric cryptography
- Block size n for a fixed small n (e.g., 128) and key size κ
- $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$
- Standard security assumption: Pseudorandom Permutation (PRP)
- Commonly used to build Pseudorandom Functions (PRFs), e.g., in Counter Mode
- Counter Mode: $z \leftarrow E_K(x\|0)\|E_K(x\|1)\| \ldots E_K(x\|i) \ldots$
- x is a nonce, z is a random bitstream
- This is a secure PRF as long as E_K is not called more than $2^{n/2}$ times
- The $2^{n/2}$ term is the "birthday bound" PRP-PRF switching term
- Can get uncomfortable small when $n = 64$ (e.g., in PRINCE, Midori, SKINNY, Simon, PRESENT)

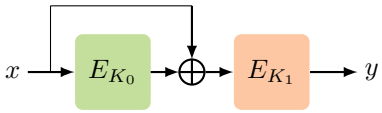
PRP-to-PRF

Truncation

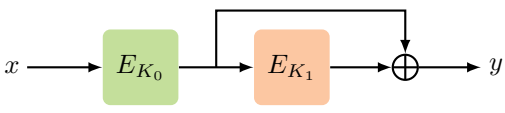


Sum Of Permutations





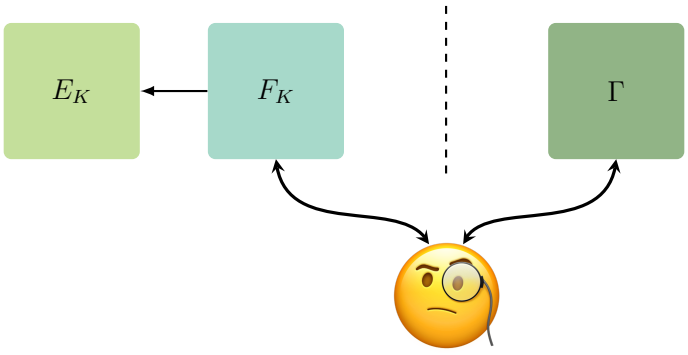
EDM



EDMD

Security Model: Indistinguishability

- F_k is a keyed function based on secret permutation Π, and Γ is a random function.



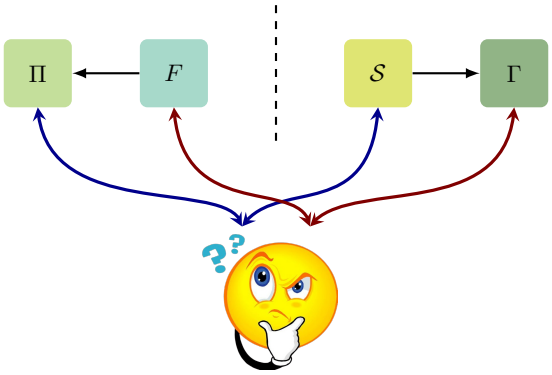
$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{F}}(\mathcal{A}) \coloneqq |\Pr_K[\mathcal{A}^{\mathsf{F}_K} \to 1] - \Pr_\Gamma[\mathcal{A}^{\Gamma} \to 1]|$$

Security in Keyless Setting

- PRF security of these schemes is well-understood by now
- However, PRF security depends heavily on the secrecy of the key
- Some level of security may also be desired when the key is leaked or even adversary-influenced
- An example is the notion of *committing security*
- Committing attack: find distinct $(K, x), (K', x')$ such that $F_K(x) = F_{K'}(x')$
- Comes in several variants, depending on application
- Encompassed by the stronger notion of *indifferentiability*
- This lets us build keyless Random Oracles from Random Permutations
- Here we focus on "short input" random oracles (an input of fixed width n)

Security Model: Indifferentiability

- F is a function based on a random permutation Π and Γ is a random oracle
- $\mathcal{S}$ is a simulator, which has access to Γ and imitates the public permutation Π



$$\mathsf{Adv}^{\mathsf{indiff}}_{F^{\Pi},\Gamma}(\mathcal{A}) := |\Pr_{\Pi}[\mathcal{A}^{F^{\Pi},\Pi} \to 1] - \Pr_{\Gamma}[\mathcal{A}^{\Gamma,\mathcal{S}^{\Gamma}} \to 1]|$$

Characterizing Short-I/O Random Oracles

- Π_0, Π_1 are two permutations over $\{0,1\}^n$
- $\alpha_i, \beta_i, \gamma_i \in \mathrm{GL}(\mathbf{F}_2^n) \cup \{0\}$ and $x \in \{0,1\}^n$:

$$\mathtt{F}^{\Pi_0,\Pi_1}(x) = \gamma_1 x \oplus \gamma_2 \Pi_0(\alpha_1 x) \oplus \gamma_3 \Pi_1(\beta_1 x \oplus \beta_2 \Pi_0(\alpha_1 x))$$

- $\mathtt{F}$ can be computed with the equation

$$\begin{pmatrix} u_0 \\ u_1 \\ \mathtt{F}(x) \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 & 0 \\ \beta_1 & \beta_2 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} \begin{pmatrix} x \\ \Pi_0(u_0) \\ \Pi_1(u_1) \end{pmatrix}$$

- We can assume $\gamma_1 = 0$, since the adversary can remove the effect of $\gamma_1 x$

Observations: Some Trivial Attacks

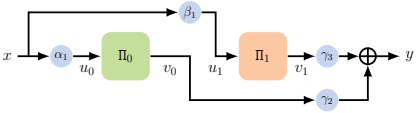
$$\mathtt{F}^{\Pi_0,\Pi_1}(x) = \gamma_2\Pi_0(\alpha_1 x) \oplus \gamma_3\Pi_1(\beta_1 x \oplus \beta_2\Pi_0(\alpha_1 x))$$

$$A = \begin{pmatrix} \alpha_1 & 0 & 0 \\ \beta_1 & \beta_2 & 0 \\ 0 & \gamma_2 & \gamma_3 \end{pmatrix}$$

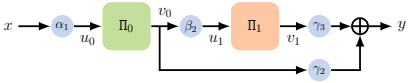
- When A contains either a zero row or a zero column, the influence of either the input x or one of Π_0 or Π_1 is lost, leading to easy attacks
- When $\beta_1 = \gamma_2 = 0$, $\mathtt{F}(x) = \gamma_3\Pi_1(\beta_2\Pi_0(\alpha_1 x))$, leading to the following attack:
  - Choose x
  - $z \leftarrow \mathtt{F}(\alpha_1^{-1}x)$
  - $v \leftarrow \Pi_1^{-1}(\gamma_3^{-1}z)$
  - $u \leftarrow \Pi_0^{-1}(\beta_2^{-1}v)$
  - Check whether the output equals x
  - In ideal world simulator cannot guess x after seeing z and v

Remaining Possible Constructions

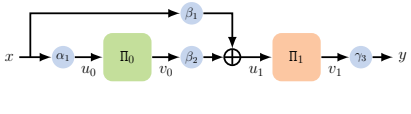
$$\mathrm{F}^{\Pi_0,\Pi_1}(x) = \gamma_2\Pi_0(\alpha_1 x) \oplus \gamma_3\Pi_1(\beta_1 x \oplus \beta_2\Pi_0(\alpha_1 x))$$



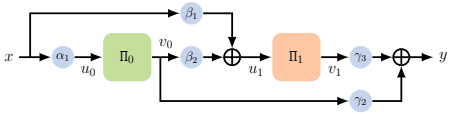
$A_{\texttt{SOP}}$ $(\beta_2 = 0, \beta_1, \gamma_2 \neq 0)$



$A_{\texttt{EDMD}}$ $(\beta_1 = 0, \beta_2, \gamma_2 \neq 0)$



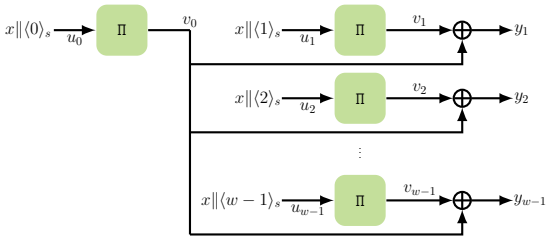
$A_{\texttt{EDM}}$ $(\gamma_2 = 0, \beta_1, \beta_2 \neq 0)$



$A_{\texttt{DMDM}}$ $(\beta_1, \beta_2, \gamma_2 \neq 0)$

Overview of analysed constructions

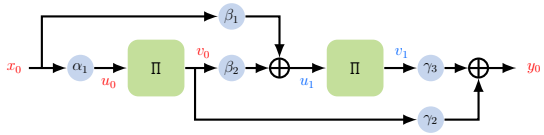
- We analyze these four constructions under three different assumptions on Π_0 and Π_1:
  - $\Pi_1 = \Pi_0$
  - $\Pi_1 = \Pi_0^{-1}$
  - Π_0 and Π_1 are independent
- In addition we look at the short-input long-output random oracle $\mathtt{XORP}^{\Pi}_{w}$:



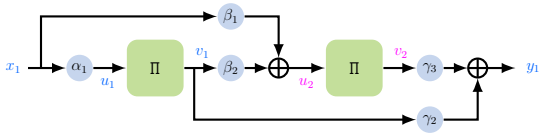
Summary of Analysis

- None of these is indifferentiable when $\Pi_1 = \Pi_0$
- None of these is indifferentiable when $\Pi_1 = \Pi_0^{-1}$
- Both these cases are broken with a new *chaining attack*
- When Π_0 and Π_1 are independent, we have positive results:
  - SOP and EDMD are indifferentiable up to $2^{2n/3}$ queries
  - EDM and DMDM are indifferentiable up to $2^{n/2}$ queries (with matching attacks)
- Further we show that $\mathtt{XORP}_w$ is indifferentiable up to $2^{2n/3}/w^{4/3}$ queries

Creating a Chain when $\Pi_1 = \Pi_0 = \Pi$



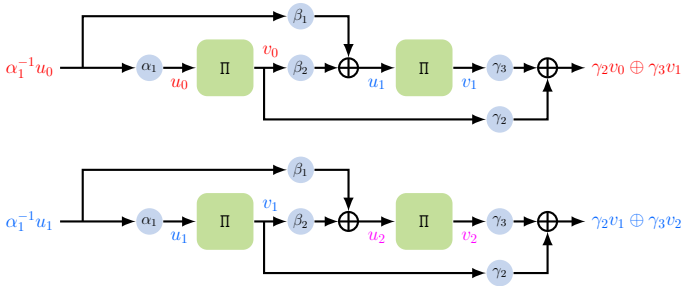
- Choose v_0
- Query $u_0 \leftarrow \Pi^{-1}(v_0)$
- $x_0 \leftarrow \alpha_1^{-1} u_0$
- Query $y_0 \leftarrow \mathtt{F}(x_0)$
- $u_1 \leftarrow \beta_1 \alpha_1^{-1} u_0 \oplus \beta_2 v_0$
- $v_1 \leftarrow \gamma_3^{-1}(y_0 \oplus \gamma_2 v_0)$



- We already have v_1 and u_1
- $x_1 \leftarrow \alpha_1^{-1} u_1$
- Query $y_1 \leftarrow \mathtt{F}(x_1)$
- $u_2 \leftarrow \beta_1 \alpha_1^{-1} u_1 \oplus \beta_2 v_1$
- $v_2 \leftarrow \gamma_3^{-1}(y_1 \oplus \gamma_2 v_1)$

Creating a Chain when $\Pi_1 = \Pi_0 = \Pi$

- $((u_0, v_0), (u_1, v_1), (u_2, v_2) \ldots, (u_m, v_m))$ is said to be an m-*chain* if $v_i = \Pi(u_i)$ and $u_{i+1} = \beta_1 \alpha_1^{-1} u_i \oplus \beta_2 v_i$ for all $i \in [m-1]$

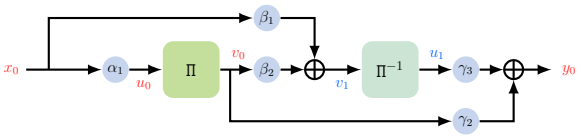


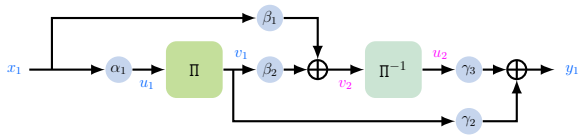
- In the real world, $\mathtt{F}(\alpha_1^{-1}u_0) \oplus \cdots \oplus \mathtt{F}(\alpha_1^{-1}u_i) = \gamma_2 v_0 \oplus \bigoplus_{j=1}^{i} (\gamma_2 \oplus \gamma_3) v_j \oplus \gamma_3 v_{i+1}$
- Efficient simulator cannot maintain this relation for large m

Mounting an Attack using a Chain

- Creating and testing a chain requires only two simulator queries
- Simulator's expected complexity is polynomial in the numer of simulator queries
- A simulator receiving two queries can make $O(2)$ queries to Γ
- For some m_{MAX}, it can keep track of a chain of length at most m_{MAX} (with high probability)
- An adversary testing a chain of length $m_{\mathsf{MAX}} + 1$ wins with high probability
- Thus for any simulator there is an adversary it cannot defeat

Creating a Chain when $\Pi_0 = \Pi, \Pi_1 = \Pi^{-1}$



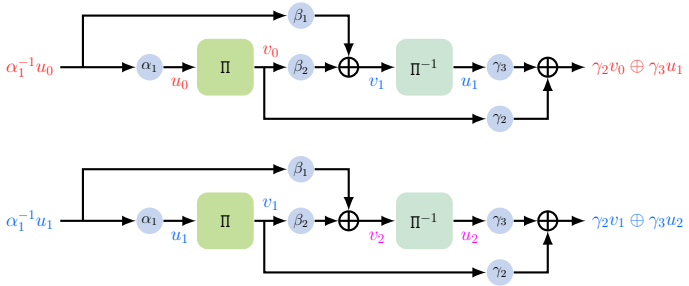


- Choose u_0
- Query $v_0 \leftarrow \Pi(u_0)$
- $x_0 \leftarrow \alpha_1^{-1} u_0$
- Query $y_0 \leftarrow \mathtt{F}(x_0)$
- $v_1 \leftarrow \beta_1 \alpha_1^{-1} u_0 \oplus \beta_2 v_0$
- $u_1 \leftarrow \gamma_3^{-1}(y_0 \oplus \gamma_2 v_0)$

- We already have u_1 and v_1
- $x_1 \leftarrow \alpha_1^{-1} u_1$
- Query $y_1 \leftarrow \mathtt{C}(x_1)$
- $v_2 \leftarrow \beta_1 \alpha_1^{-1} u_1 \oplus \beta_2 v_1$
- $u_2 \leftarrow \gamma_3^{-1}(y_1 \oplus \gamma_2 v_1)$

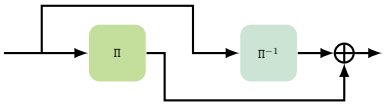
Creating a Chain when $\Pi_0 = \Pi, \Pi_1 = \Pi^{-1}$

- $((u_0, v_0), (u_1, v_1), (u_2, v_2) \ldots, (u_m, v_m))$ is said to be an m-chain if $v_i = \Pi(u_i)$ and $u_{i+1} = \gamma_3^{-1}(\mathtt{F}(\alpha^{-1}u_i) \oplus \gamma_2 v_i)$ for all $i \in [m-1]$



- In the real world, $\mathtt{F}(\alpha_1^{-1}u_0) \oplus \cdots \oplus \mathtt{F}(\alpha_1^{-1}u_i) = \bigoplus_{j=0}^{i} \gamma_2 v_j \oplus \bigoplus_{j=0}^{i} \gamma_3 u_{j+1}$

An Interesting Corollary



- Dodis et al. proposed [DPP08, Lemma 4] a simulator for this construction
- They claimed a birthday-bound (strong) indifferentiability security with the proposed simulator
- However, our chaining attack invalidates this claimed result
- We show a concrete attack on the proposed simulator
- More generally, we show that no simulator can work for this construction

[DPP08] Dodis, Y., Pietrzak, K., Puniya, P.: A New Mode of Operation for Block Ciphers and Length-Preserving MACs. EUROCRYPT 2008

Applications

- Indifferentiability can be translated to committing security by considering a multi-instance setting
- Constructions using hash function are often proved secure in the Random Oracle model
- Indifferentiability results justify instantiating a Random Oracle with a concrete construction
- Short Input ROs are used in the Fiat-Shamir transform to make public-coin proofs non-interactive
- MPC-in-the-head protocols can use $\texttt{XORP}_w$ instead of GGM trees for better efficiency
- Many proofs of public-key protocols rely on Short Input ROs
- Our overall impression: having a suite of Short Input ROs should come in handy

Thank You








https://ia.cr/2026/336