On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction

Avradip Mandal[1] Jacques Patarin[2] **Yannick Seurin**[3]

[1]University of Luxembourg

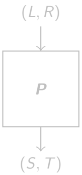
[2]University of Versailles, France

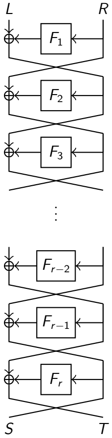
[3]ANSSI, France

March 20, TCC 2012

Context

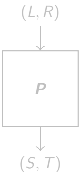
- building cryptographic permutations from cryptographic functions: the r-round Feistel construction Ψ_r
- round functions = random oracles $\boldsymbol{F}$
- does the Feistel construction $\Psi_r^{\boldsymbol{F}}$ "behave" as a random permutation $\boldsymbol{P}$?
- secret round functions
  $\Rightarrow$ Luby-Rackoff
- public round functions
  $\Rightarrow$ indifferentiability framework [MRH04]

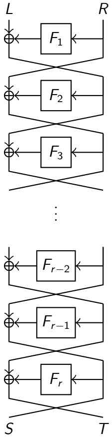




Context

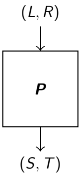
- building cryptographic permutations from cryptographic functions: the r-round Feistel construction Ψ_r
- round functions = random oracles $\boldsymbol{F}$
- does the Feistel construction $\Psi_r^{\boldsymbol{F}}$ "behave" as a random permutation $\boldsymbol{P}$?
- secret round functions
  $\Rightarrow$ Luby-Rackoff
- public round functions
  $\Rightarrow$ indifferentiability framework [MRH04]

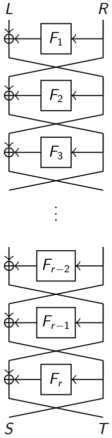




Context

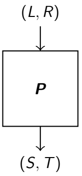
- building cryptographic permutations from cryptographic functions: the r-round Feistel construction Ψ_r
- round functions = random oracles $\boldsymbol{F}$
- does the Feistel construction $\Psi_r^{\boldsymbol{F}}$ "behave" as a random permutation $\boldsymbol{P}$?
- secret round functions
  $\Rightarrow$ Luby-Rackoff
- public round functions
  $\Rightarrow$ indifferentiability framework [MRH04]

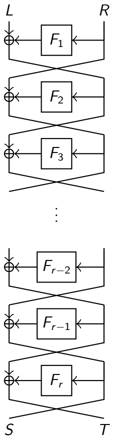




Context

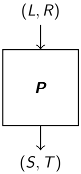
- building cryptographic permutations from cryptographic functions: the r-round Feistel construction Ψ_r
- round functions = random oracles $\boldsymbol{F}$
- does the Feistel construction $\Psi_r^{\boldsymbol{F}}$ "behave" as a random permutation $\boldsymbol{P}$?
- secret round functions
  $\Rightarrow$ Luby-Rackoff
- public round functions
  $\Rightarrow$ indifferentiability framework [MRH04]

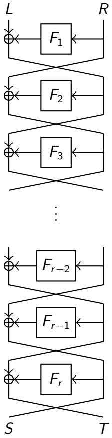




Context

- building cryptographic permutations from cryptographic functions: the r-round Feistel construction Ψ_r
- round functions = random oracles $\boldsymbol{F}$
- does the Feistel construction $\Psi_r^{\boldsymbol{F}}$ "behave" as a random permutation $\boldsymbol{P}$?
- secret round functions
  $\Rightarrow$ Luby-Rackoff
- public round functions
  $\Rightarrow$ indifferentiability framework [MRH04]





In this talk

- we consider weaker notions of indifferentiability:
  - public indifferentiability
  - sequential indifferentiability

  and show them to be equivalent
- we show that the Feistel construction with 6 rounds is publicly indifferentiable from a random permutation (14 rounds best known result for full indifferentiability [HKT11])
- we link the notion of public indifferentiability with the notion of correlation intractability of [CGH98]

In this talk

- we consider weaker notions of indifferentiability:
  - public indifferentiability
  - sequential indifferentiability

  and show them to be equivalent
- we show that the Feistel construction with 6 rounds is publicly indifferentiable from a random permutation (14 rounds best known result for full indifferentiability [HKT11])
- we link the notion of public indifferentiability with the notion of correlation intractability of [CGH98]

In this talk

- we consider weaker notions of indifferentiability:
  - public indifferentiability
  - sequential indifferentiability

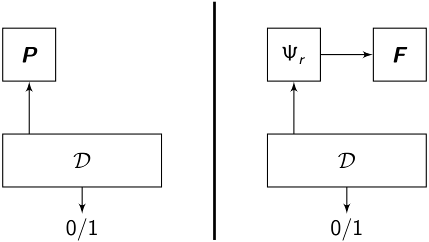
  and show them to be equivalent
- we show that the Feistel construction with 6 rounds is publicly indifferentiable from a random permutation (14 rounds best known result for full indifferentiability [HKT11])
- we link the notion of public indifferentiability with the notion of correlation intractability of [CGH98]

Outline

1. Public and Sequential Indifferentiability
2. Public Indifferentiability of the 6-Round Feistel Construction
3. Correlation Intractability

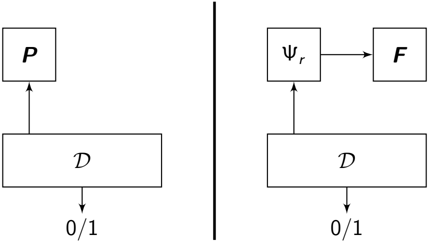
Outline

The classical indistinguishability notion



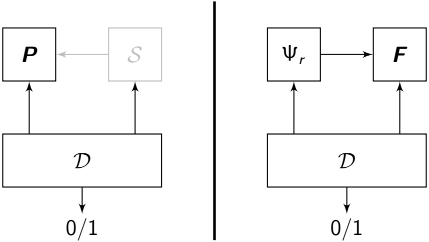
- the distinguisher cannot access the round functions.
- Luby-Rackoff theorem: Ψ_3 is indist. from a random permutation, Ψ_4 is indist. from an invertible random permutation

The classical indistinguishability notion



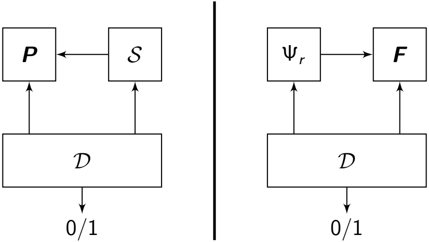
- the distinguisher cannot access the round functions.
- Luby-Rackoff theorem: Ψ_3 is indist. from a random permutation, Ψ_4 is indist. from an invertible random permutation

Full indifferentiability



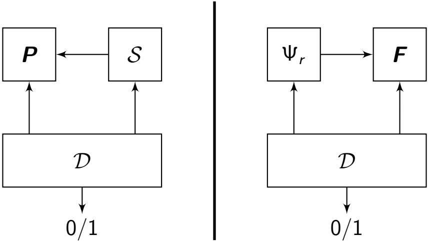
- Ψ_r^F is indifferentiable from P is there exists an (efficient) simulator $\mathcal{S}$ such that $(P, \mathcal{S}^P)$ and (Ψ_r^F, F) are indist.
- the simulator does not know $\mathcal{D}$'s queries to P
- best known result for Feistel: 14 rounds [HKT11]

Full indifferentiability



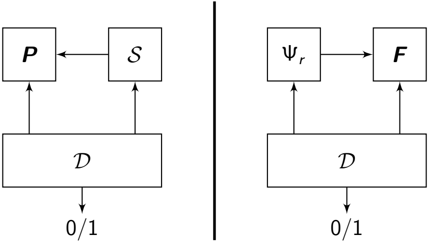
- $\Psi_r^{\boldsymbol{F}}$ is indifferentiable from $\boldsymbol{P}$ is there exists an (efficient) simulator $\mathcal{S}$ such that $(\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$ and $(\Psi_r^{\boldsymbol{F}}, \boldsymbol{F})$ are indist.
- the simulator does not know $\mathcal{D}$'s queries to $\boldsymbol{P}$
- best known result for Feistel: 14 rounds [HKT11]

Full indifferentiability



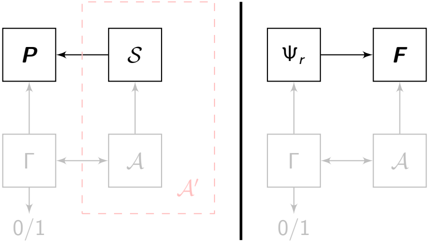
- $\Psi_r^{\boldsymbol{F}}$ is indifferentiable from $\boldsymbol{P}$ is there exists an (efficient) simulator $\mathcal{S}$ such that $(\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$ and $(\Psi_r^{\boldsymbol{F}}, \boldsymbol{F})$ are indist.
- the simulator does not know $\mathcal{D}$'s queries to $\boldsymbol{P}$
- best known result for Feistel: 14 rounds [HKT11]

Full indifferentiability



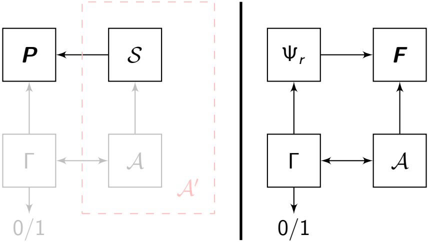
- $\Psi_r^{\boldsymbol{F}}$ is indifferentiable from $\boldsymbol{P}$ is there exists an (efficient) simulator $\mathcal{S}$ such that $(\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$ and $(\Psi_r^{\boldsymbol{F}}, \boldsymbol{F})$ are indist.
- the simulator does not know $\mathcal{D}$'s queries to $\boldsymbol{P}$
- best known result for Feistel: 14 rounds [HKT11]

Composition theorem



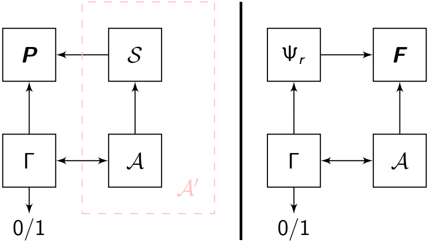
- an attacker $\mathcal{A}$ against cryptosystem Γ used with $\Psi_r^{\boldsymbol{F}}$...
- ...implies an attacker $\mathcal{A}'$ against Γ used with $\boldsymbol{P}$
- true for single-stage security games only [RSS11]

Composition theorem



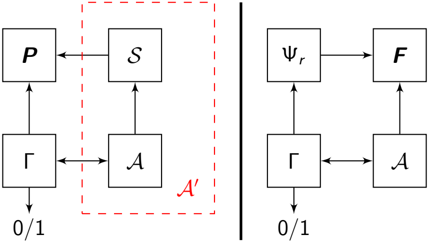
- an attacker $\mathcal{A}$ against cryptosystem Γ used with $\Psi_r^{\boldsymbol{F}}$...
- ...implies an attacker $\mathcal{A}'$ against Γ used with $\boldsymbol{P}$
- true for single-stage security games only [RSS11]

Composition theorem



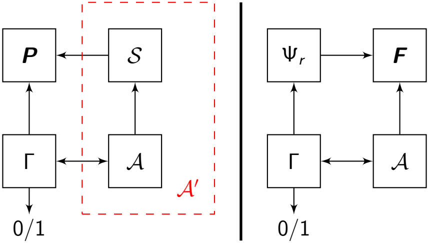
- an attacker $\mathcal{A}$ against cryptosystem Γ used with $\Psi_r^{\boldsymbol{F}}$...
- ...implies an attacker $\mathcal{A}'$ against Γ used with $\boldsymbol{P}$
- true for single-stage security games only [RSS11]

Composition theorem



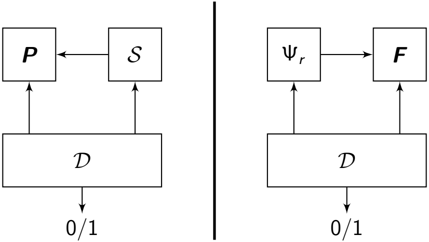
- an attacker $\mathcal{A}$ against cryptosystem Γ used with $\Psi_r^{\boldsymbol{F}}$...
- ...implies an attacker $\mathcal{A}'$ against Γ used with $\boldsymbol{P}$
- true for single-stage security games only [RSS11]

Composition theorem



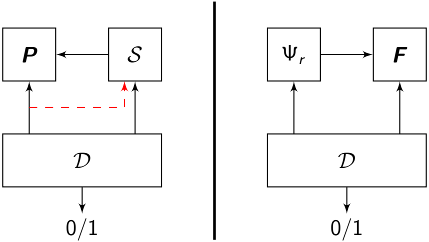
- an attacker $\mathcal{A}$ against cryptosystem Γ used with $\Psi_r^{\boldsymbol{F}}$...
- ...implies an attacker $\mathcal{A}'$ against Γ used with $\boldsymbol{P}$
- true for single-stage security games only [RSS11]

Public indifferentiability [YMO09,DRS09]



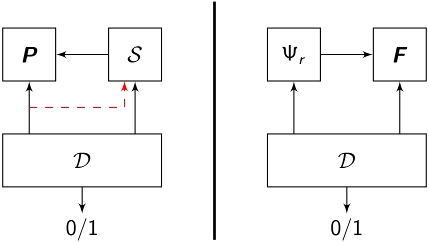
- weaker notion where the simulator is given all queries made by $\mathcal{D}$ to $\boldsymbol{P}$
- composition theorem still holds for cryptosystems where all queries to $\boldsymbol{P}$ can be revealed to the adversary without affecting security (*e.g.* "hash-and-sign" signature schemes)

Public indifferentiability [YMO09,DRS09]



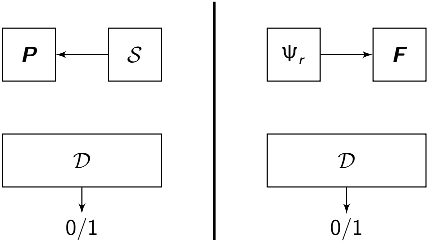
- weaker notion where the simulator is given all queries made by $\mathcal{D}$ to $\boldsymbol{P}$
- composition theorem still holds for cryptosystems where all queries to $\boldsymbol{P}$ can be revealed to the adversary without affecting security (*e.g.* "hash-and-sign" signature schemes)

Public indifferentiability [YMO09,DRS09]



- weaker notion where the simulator is given all queries made by $\mathcal{D}$ to $\boldsymbol{P}$
- composition theorem still holds for cryptosystems where all queries to $\boldsymbol{P}$ can be revealed to the adversary without affecting security (*e.g.* "hash-and-sign" signature schemes)

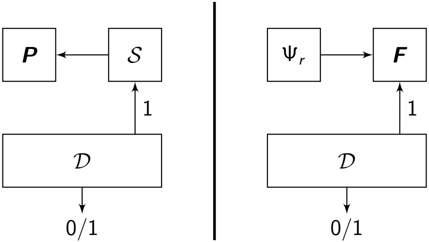
Sequential indifferentiability



The distinguisher can:

1. query $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ in a first phase
2. query $\boldsymbol{P}/\Psi_r^{\boldsymbol{F}}$ in a second phase, but not $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ any more
3. not intrinsically interesting, tool to prove public indiff.

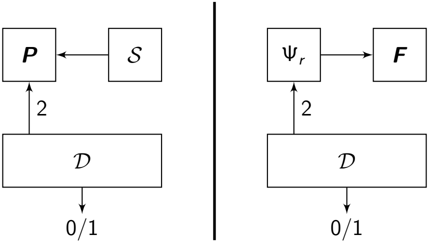
Sequential indifferentiability



The distinguisher can:

1. query $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ in a first phase
2. query $\boldsymbol{P}/\Psi_r^{\boldsymbol{F}}$ in a second phase, but not $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ any more
3. not intrinsically interesting, tool to prove public indiff.

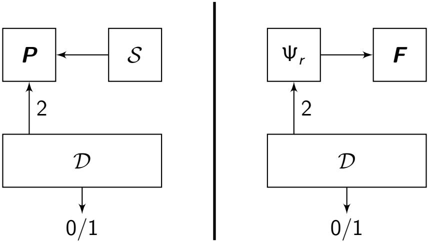
Sequential indifferentiability



The distinguisher can:

1. query $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ in a first phase
2. query $\boldsymbol{P}/\Psi_r^{\boldsymbol{F}}$ in a second phase, but not $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ any more
3. not intrinsically interesting, tool to prove public indiff.

Sequential indifferentiability



The distinguisher can:

1. query $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ in a first phase
2. query $\boldsymbol{P}/\Psi_r^{\boldsymbol{F}}$ in a second phase, but not $\mathcal{S}^{\boldsymbol{P}}/\boldsymbol{F}$ any more
3. not intrinsically interesting, tool to prove public indiff.

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious
  (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious
  (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious
  (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious
  (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Seq-indiff. $\Leftrightarrow$ Pub-indiff. (for stateless primitives $\boldsymbol{P}$)

- $\boldsymbol{P}$ is stateless = its answers are independent of the order of queries it receives
- NB: an invertible random permutation is stateless
- pub-indiff $\Rightarrow$ seq-indiff: obvious
  (in the seq-indiff. game, the simulator is done once the distinguisher makes its first query to $\boldsymbol{P}$)
- seq-indiff $\Rightarrow$ pub-indiff for stateless ideal primitives $\boldsymbol{P}$
- idea of the proof: starting from a simulator $\mathcal{S}_{\mathrm{seq}}$ for seq-indiff., one builds a simulator $\mathcal{S}_{\mathrm{pub}}$ which emulates all queries of the distinguisher to $\boldsymbol{P}$ by calling $\Psi_r^{\mathcal{S}_{\mathrm{seq}}^{\boldsymbol{P}}}$.
- counterexample (in the computational case) when $\boldsymbol{P}$ is stateful [Ristenpart]

Outline

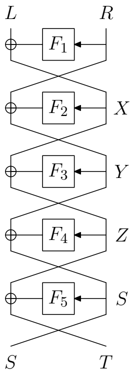
1. Public and Sequential Indifferentiability
2. Public Indifferentiability of the 6-Round Feistel Construction
3. Correlation Intractability

5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



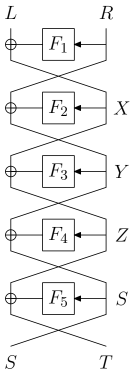
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential

X_{12}



5 rounds are not enough for seq/pub-indifferentiability

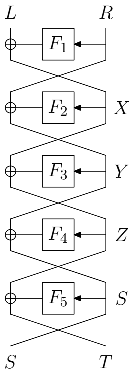
- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential

X_{12}

Y_{14} Y_{23}

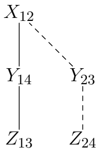


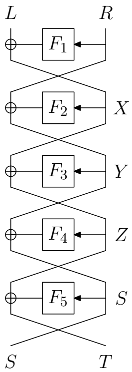
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



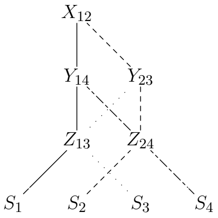


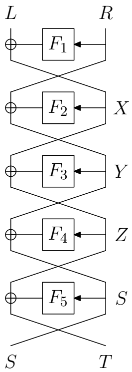
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



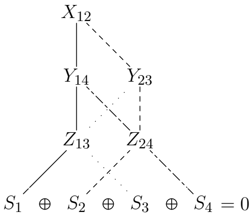


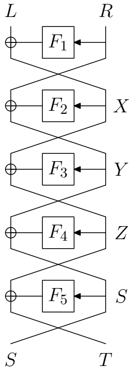
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with ***P***
- the distinguisher is sequential



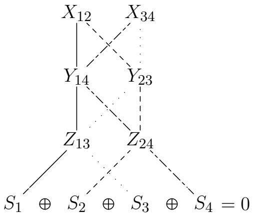


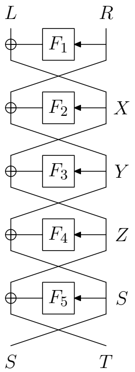
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



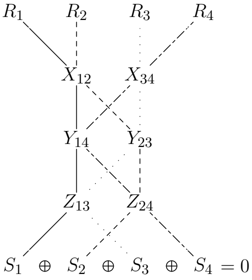


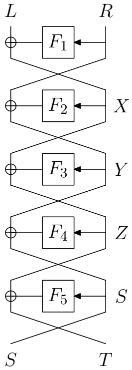
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



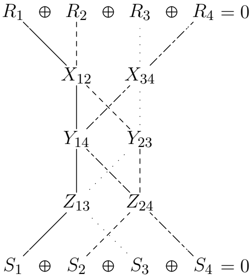


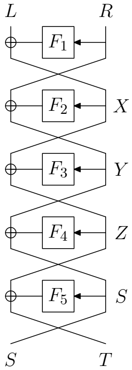
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



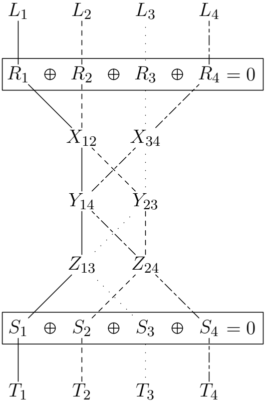


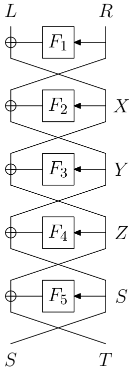
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



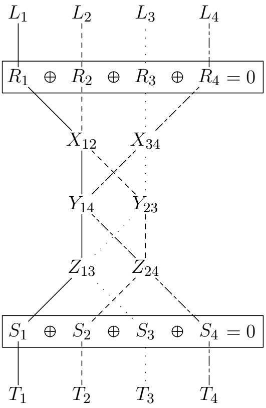


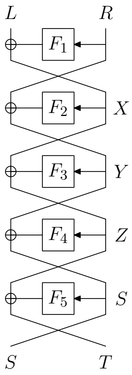
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



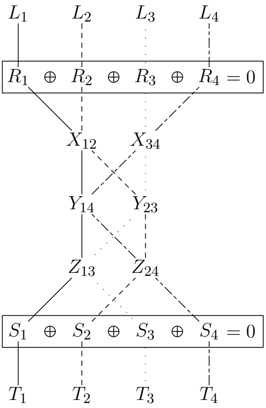


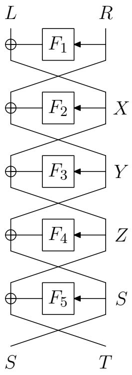
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



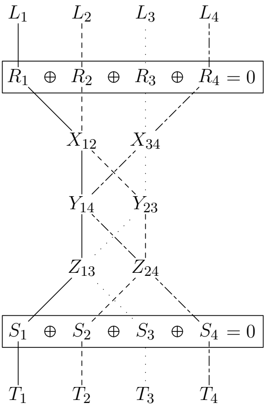


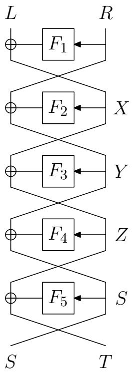
5 rounds are not enough for seq/pub-indifferentiability

- For Ψ_5, it is possible to find four inputs /outputs such that

$$\begin{cases} R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0 \\ S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0 \end{cases}$$

- impossible for a random permutation
- $\Rightarrow$ the simulator cannot be coherent with $\boldsymbol{P}$
- the distinguisher is sequential



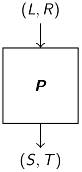


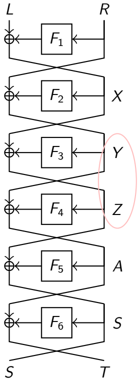
Simulation strategy for 6 rounds

- the simulator must return answers:
  - coherent with $\boldsymbol{P}$:

$$\forall L, R, \ \Psi_6(L, R) = \boldsymbol{P}(L, R)$$

  - indist. from uniformly random
- the simulator maintains an history of answers for each F_i
- it completes in advance the Feistel for all centers $(Y, Z) \in F_3 \times F_4$ in the history, adapting some round function values to match the random permutation



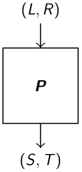


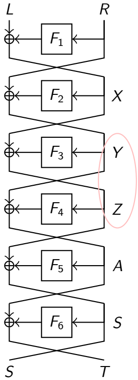
Simulation strategy for 6 rounds

- the simulator must return answers:
  - coherent with $\boldsymbol{P}$:

$$\forall L, R,\ \Psi_6(L, R) = \boldsymbol{P}(L, R)$$

  - indist. from uniformly random
- the simulator maintains an history of answers for each F_i
- it completes in advance the Feistel for all centers $(Y, Z) \in F_3 \times F_4$ in the history, adapting some round function values to match the random permutation



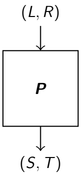


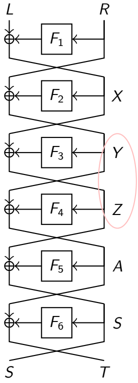
Simulation strategy for 6 rounds

- the simulator must return answers:
  - coherent with $\boldsymbol{P}$:

$$\forall L, R,\ \Psi_6(L, R) = \boldsymbol{P}(L, R)$$

  - indist. from uniformly random
- the simulator maintains an history of answers for each F_i
- it completes in advance the Feistel for all centers $(Y, Z) \in F_3 \times F_4$ in the history, adapting some round function values to match the random permutation



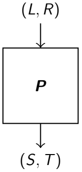


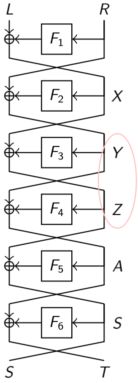
Simulation strategy for 6 rounds

- the simulator must return answers:
  - coherent with $\boldsymbol{P}$:

$$\forall L, R,\ \Psi_6(L, R) = \boldsymbol{P}(L, R)$$

  - indist. from uniformly random
- the simulator maintains an history of answers for each F_i
- it completes in advance the Feistel for all centers $(Y, Z) \in F_3 \times F_4$ in the history, adapting some round function values to match the random permutation



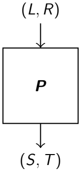


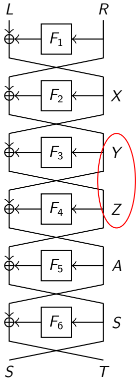
Simulation strategy for 6 rounds

- the simulator must return answers:
  - coherent with $\boldsymbol{P}$:

$$\forall L, R,\ \Psi_6(L, R) = \boldsymbol{P}(L, R)$$

  - indist. from uniformly random
- the simulator maintains an history of answers for each F_i
- it completes in advance the Feistel for all centers $(Y, Z) \in F_3 \times F_4$ in the history, adapting some round function values to match the random permutation





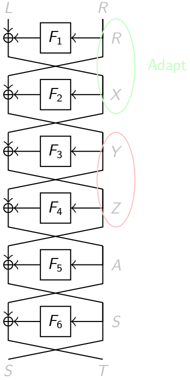
Completing centers

When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$



Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$

Completing centers

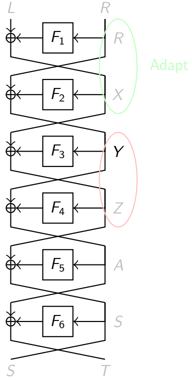
When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

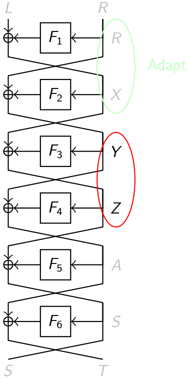
When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

When receiving a query for $F_3(Y)$, the simulator:

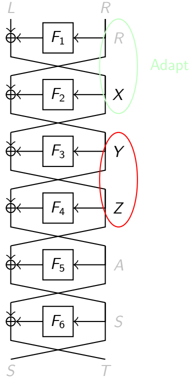
- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$

$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

When receiving a query for $F_3(Y)$, the simulator:

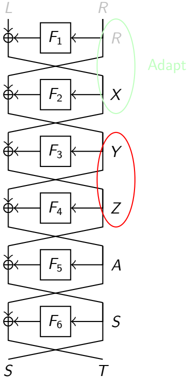
- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$

$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

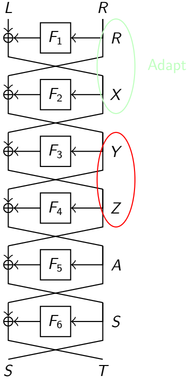
When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

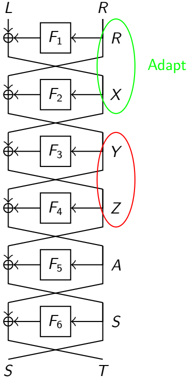
When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Completing centers

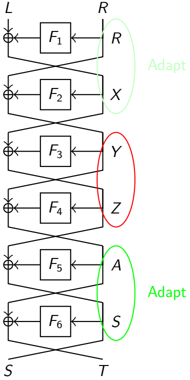
When receiving a query for $F_3(Y)$, the simulator:

- sets $F_3(Y)$ unif. at random
- for all $Z \in F_4$, it completes the chain (Y, Z):
  - compute $X = Z \oplus F_3(Y)$
  - compute A, S, T
  - query $(L, R) = \boldsymbol{P}^{-1}(S, T)$
  - adapt $F_1(R)$ and $F_2(X)$:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$

so that $\Psi_6(L, R) = \boldsymbol{P}(L, R)$

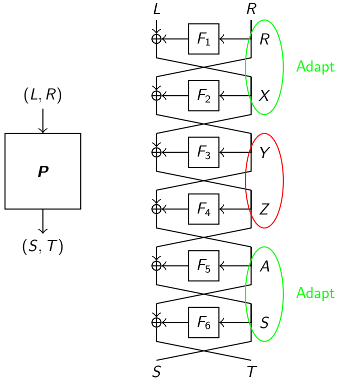
Symmetric for a query $F_4(Z)$
$\rightarrow$ adapt $F_5(A)$ and $F_6(S)$



Indifferentiability proof

Two main points in the indifferentiability proof:

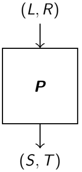
1. the simulator is polynomial-time
2. the simulator can always adapt round function values $(F_1(R), F_2(X))$ or $(F_5(A), F_6(S))$

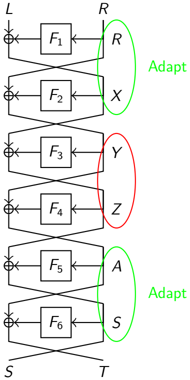


Indifferentiability proof

Two main points in the indifferentiability proof:

1. the simulator is polynomial-time
2. the simulator can always adapt round function values $(F_1(R), F_2(X))$ or $(F_5(A), F_6(S))$

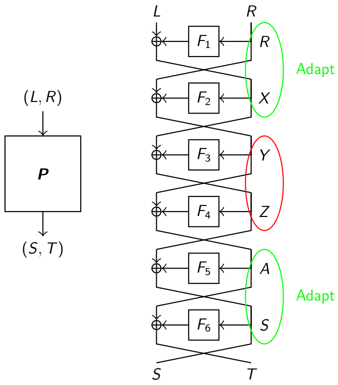




Indifferentiability proof

Two main points in the indifferentiability proof:

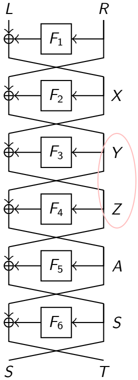
1. the simulator is polynomial-time
2. the simulator can always adapt round function values $(F_1(R), F_2(X))$ or $(F_5(A), F_6(S))$



The simulator is polynomial-time

If the distinguisher makes at most q queries, then:

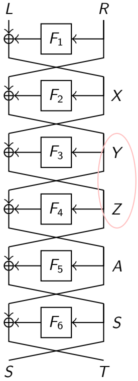
- the size of history of F_3 and F_4 is at most q
- the simulator completes at most q^2 centers (Y, Z)
- the size of history of F_1, F_2, F_5, F_6 is at most $q^2 + q$
- the simulator makes at most q^2 queries to $\boldsymbol{P}$



The simulator is polynomial-time

If the distinguisher makes at most q queries, then:

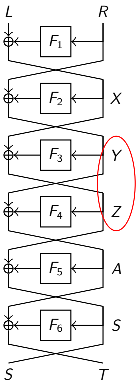
- the size of history of F_3 and F_4 is at most q
- the simulator completes at most q^2 centers (Y, Z)
- the size of history of F_1, F_2, F_5, F_6 is at most $q^2 + q$
- the simulator makes at most q^2 queries to $\boldsymbol{P}$



The simulator is polynomial-time

If the distinguisher makes at most q queries, then:

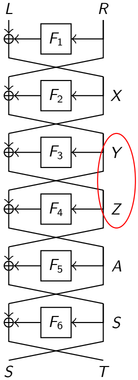
- the size of history of F_3 and F_4 is at most q
- the simulator completes at most q^2 centers (Y, Z)
- the size of history of F_1, F_2, F_5, F_6 is at most $q^2 + q$
- the simulator makes at most q^2 queries to $\boldsymbol{P}$



The simulator is polynomial-time

If the distinguisher makes at most q queries, then:

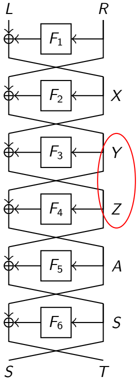
- the size of history of F_3 and F_4 is at most q
- the simulator completes at most q^2 centers (Y, Z)
- the size of history of F_1, F_2, F_5, F_6 is at most $q^2 + q$
- the simulator makes at most q^2 queries to $\boldsymbol{P}$



The simulator is polynomial-time

If the distinguisher makes at most q queries, then:

- the size of history of F_3 and F_4 is at most q
- the simulator completes at most q^2 centers (Y, Z)
- the size of history of F_1, F_2, F_5, F_6 is at most $q^2 + q$
- the simulator makes at most q^2 queries to $\boldsymbol{P}$



The simulator can always adapt

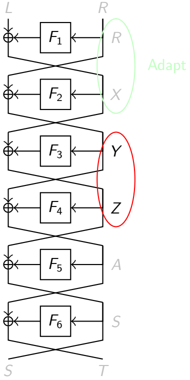
When completing a center (Y, Z) after a query for $F_3(Y)$:

- $X = Z \oplus F_3(Y)$, where $F_3(Y)$ is unif. random
  $\Rightarrow X \in F_2$ with negl. probability only
- (L, R) are obtained by querying

$$(L, R) = \boldsymbol{P}^{-1}(S, T)$$

  $\Rightarrow L$ and R are close to unif. random
  $\Rightarrow R \in F_1$ with negl. probability only
- $F_1(R)$ and $F_2(X)$ are close to unif. random:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$



The simulator can always adapt

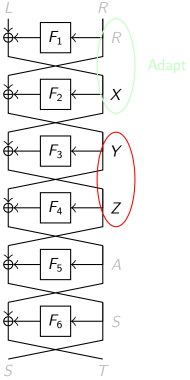
When completing a center (Y, Z) after a query for $F_3(Y)$:

- $X = Z \oplus F_3(Y)$, where $F_3(Y)$ is unif. random
  $\Rightarrow X \in F_2$ with negl. probability only
- (L, R) are obtained by querying

$$(L, R) = \boldsymbol{P}^{-1}(S, T)$$

  $\Rightarrow$ L and R are close to unif. random
  $\Rightarrow$ $R \in F_1$ with negl. probability only
- $F_1(R)$ and $F_2(X)$ are close to unif. random:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$



The simulator can always adapt

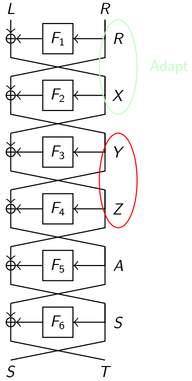
When completing a center (Y, Z) after a query for $F_3(Y)$:

- $X = Z \oplus F_3(Y)$, where $F_3(Y)$ is unif. random
  $\Rightarrow X \in F_2$ with negl. probability only
- (L, R) are obtained by querying

$$(L, R) = \boldsymbol{P}^{-1}(S, T)$$

  $\Rightarrow L$ and R are close to unif. random
  $\Rightarrow R \in F_1$ with negl. probability only
- $F_1(R)$ and $F_2(X)$ are close to unif. random:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$



The simulator can always adapt

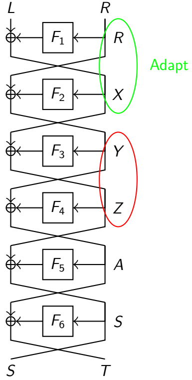
When completing a center (Y, Z) after a query for $F_3(Y)$:

- $X = Z \oplus F_3(Y)$, where $F_3(Y)$ is unif. random
  $\Rightarrow X \in F_2$ with negl. probability only
- (L, R) are obtained by querying

$$(L, R) = \boldsymbol{P}^{-1}(S, T)$$

  $\Rightarrow L$ and R are close to unif. random
  $\Rightarrow R \in F_1$ with negl. probability only
- $F_1(R)$ and $F_2(X)$ are close to unif. random:

$$\begin{cases} F_1(R) = L \oplus X \\ F_2(X) = R \oplus Y \end{cases}$$



Outline

Evasive relation

Definition (Evasive relation)

A relation $\mathcal{R}$ is *evasive* for ideal primitive $\boldsymbol{P}$ if it is hard, given BB access to $\boldsymbol{P}$, to find inputs $(x_1, \ldots, x_m)$ such that

$$((x_1, \ldots, x_m), (\boldsymbol{P}(x_1), \ldots, \boldsymbol{P}(x_m)) \in \mathcal{R} \ .$$

Exemple:

$$\mathcal{R} = \{((L \| 0^n), (S \| 0^n)) \ : \ L \in \{0,1\}^n, S \in \{0,1\}^n\}$$

is evasive for a $2n$-bit invertible random permutation.

Evasive relation

Definition (Evasive relation)

A relation $\mathcal{R}$ is *evasive* for ideal primitive $\boldsymbol{P}$ if it is hard, given BB access to $\boldsymbol{P}$, to find inputs $(x_1, \dots, x_m)$ such that

$$((x_1, \dots, x_m), (\boldsymbol{P}(x_1), \dots, \boldsymbol{P}(x_m)) \in \mathcal{R} \ .$$

Exemple:

$$\mathcal{R} = \{((L \| 0^n), (S \| 0^n)) \, : \, L \in \{0,1\}^n, S \in \{0,1\}^n\}$$

is evasive for a $2n$-bit invertible random permutation.

Correlation intractable construction

Definition

The construction $\Psi_r^{\boldsymbol{F}}$ is correlation intractable if for any evasive relation $\mathcal{R}$, it is hard, given BB access to $\boldsymbol{F}$, to find inputs $(x_1, \ldots, x_m)$ such that

$$((x_1, \ldots, x_m), (\Psi_r^{\boldsymbol{F}}(x_1), \ldots, \Psi_r^{\boldsymbol{F}}(x_m)) \in \mathcal{R} \ .$$

- analogous to the corresponding notion defined by [CGH98] in the standard model
- escapes impossibility results since the "key" $\boldsymbol{F}$ is exponentially long

Correlation intractable construction

Definition

The construction $\Psi_r^{\boldsymbol{F}}$ is correlation intractable if for any evasive relation $\mathcal{R}$, it is hard, given BB access to $\boldsymbol{F}$, to find inputs $(x_1, \ldots, x_m)$ such that

$$((x_1, \ldots, x_m), (\Psi_r^{\boldsymbol{F}}(x_1), \ldots, \Psi_r^{\boldsymbol{F}}(x_m)) \in \mathcal{R} \ .$$

- analogous to the corresponding notion defined by [CGH98] in the standard model
- escapes impossibility results since the “key” $\boldsymbol{F}$ is exponentially long

Correlation intractable construction

Definition

The construction $\Psi_r^{\boldsymbol{F}}$ is correlation intractable if for any evasive relation $\mathcal{R}$, it is hard, given BB access to $\boldsymbol{F}$, to find inputs $(x_1, \ldots, x_m)$ such that

$$((x_1, \ldots, x_m), (\Psi_r^{\boldsymbol{F}}(x_1), \ldots, \Psi_r^{\boldsymbol{F}}(x_m)) \in \mathcal{R} \ .$$

- analogous to the corresponding notion defined by [CGH98] in the standard model
- escapes impossibility results since the "key" $\boldsymbol{F}$ is exponentially long

Public indiff. implies correlation intractability

Theorem

If $\Psi_r^{\boldsymbol{F}}$ is pub-indiff. from $\boldsymbol{P}$, then it is correlation intractable.

The converse does not hold.

Corollary

The 6-round Feistel construction yields a correlation intractable permutation.

NB: this implies that full indiff. for 6 rounds cannot be disproved similarly to the 5-round case (by finding an evasive relation).

Public indiff. implies correlation intractability

Theorem

If $\Psi_r^{\boldsymbol{F}}$ is pub-indiff. from $\boldsymbol{P}$, then it is correlation intractable.

The converse does not hold.

Corollary

The 6-round Feistel construction yields a correlation intractable permutation.

NB: this implies that full indiff. for 6 rounds cannot be disproved similarly to the 5-round case (by finding an evasive relation).

Public indiff. implies correlation intractability

Theorem

If $\Psi_r^{\boldsymbol{F}}$ is pub-indiff. from $\boldsymbol{P}$, then it is correlation intractable.

The converse does not hold.

Corollary

The 6-round Feistel construction yields a correlation intractable permutation.

NB: this implies that full indiff. for 6 rounds cannot be disproved similarly to the 5-round case (by finding an evasive relation).

Conclusion

Sec. Notion	# Feistel rounds
PRP	3
SPRP	4
Correlation intract.	6
Public indiff.	6
Full indiff.	$6 \leq r \leq 14$

Open questions:

- minimal number of rounds for full indifferentiability?
- weaker assumptions for the round functions?
- application of seq-indiff. to hash function constructions

Conclusion

Sec. Notion	# Feistel rounds
PRP	3
SPRP	4
Correlation intract.	6
Public indiff.	6
Full indiff.	$6 \leq r \leq 14$

Open questions:

- minimal number of rounds for full indifferentiability?
- weaker assumptions for the round functions?
- application of seq-indiff. to hash function constructions

Conclusion

Sec. Notion	# Feistel rounds
PRP	3
SPRP	4
Correlation intract.	6
Public indiff.	6
Full indiff.	$6 \leq r \leq 14$

Open questions:

- minimal number of rounds for full indifferentiability?
- weaker assumptions for the round functions?
- application of seq-indiff. to hash function constructions

Conclusion

Sec. Notion	# Feistel rounds
PRP	3
SPRP	4
Correlation intract.	6
Public indiff.	6
Full indiff.	$6 \le r \le 14$

Open questions:

- minimal number of rounds for full indifferentiability?
- weaker assumptions for the round functions?
- application of seq-indiff. to hash function constructions

Conclusion

Sec. Notion	# Feistel rounds
PRP	3
SPRP	4
Correlation intract.	6
Public indiff.	6
Full indiff.	$6 \leq r \leq 14$

Open questions:

- minimal number of rounds for full indifferentiability?
- weaker assumptions for the round functions?
- application of seq-indiff. to hash function constructions

The end...

Thanks for your attention!

Comments or questions?