Foundation of Cryptography, Lecture 4
Pseudorandom Functions.

Benny Applebaum & Iftach Haitner, Tel Aviv University

Tel Aviv University.

December 1, 2016

Section 1

Informal Discussion

Motivation discussion

1. We've seen a small set of objects: $\{G(x)\}_{x\in\{0,1\}^n}$, that "looks like" a larger set of objects: $\{x\}_{x\in\{0,1\}^{2n}}$.

Motivation discussion

1. We've seen a small set of objects: $\{G(x)\}_{x \in \{0,1\}^n}$, that "looks like" a larger set of objects: $\{x\}_{x \in \{0,1\}^{2n}}$.
2. We want small set of objects: *efficient function families*, that looks like a huge set of objects: *the set of <u>all</u> functions*.

Motivation discussion

1. We've seen a small set of objects: $\{G(x)\}_{x \in \{0,1\}^n}$, that "looks like" a larger set of objects: $\{x\}_{x \in \{0,1\}^{2n}}$.
2. We want small set of objects: *efficient function families*, that looks like a huge set of objects: *the set of <u>all</u> functions*.

But

Motivation discussion

1. We've seen a small set of objects: $\{G(x)\}_{x\in\{0,1\}^n}$, that "looks like" a larger set of objects: $\{x\}_{x\in\{0,1\}^{2n}}$.
2. We want small set of objects: *efficient function families*, that looks like a huge set of objects: *the set of all functions*.

Solution

Subsection 1

Function Families

Function families

1. $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$, where $\mathcal{F}_n = \{f\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$

Function families

1. $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$, where $\mathcal{F}_n = \{f \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
2. We write $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$

Function families

1. $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$, where $\mathcal{F}_n = \{f\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
2. We write $\mathcal{F} = \{\mathcal{F}_n\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
3. If $m(n) = \ell(n) = n$, we omit it from the notation

Function families

1. $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$, where $\mathcal{F}_n = \{f\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
2. We write $\mathcal{F} = \{\mathcal{F}_n\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
3. If $m(n) = \ell(n) = n$, we omit it from the notation
4. We identify function with their description

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a “random access" source of randomness

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a “random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a "random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot
- How long does it take to describe $\pi \in \Pi_n$?

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a “random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot
- How long does it take to describe $\pi \in \Pi_n$?

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a “random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot
- How long does it take to describe $\pi \in \Pi_n$? $2^n \cdot n$ bits

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a “random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot

- How long does it take to describe $\pi \in \Pi_n$? $2^n \cdot n$ bits
- The truth table of $\pi \leftarrow \Pi_n$ is a uniform string of length $2^n \cdot n$

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- $\pi \leftarrow \Pi_n$ is a "random access" source of randomness
- Parties with access to a common $\pi \leftarrow \Pi_n$ can do a lot

- How long does it take to describe $\pi \in \Pi_n$? $2^n \cdot n$ bits
- The truth table of $\pi \leftarrow \Pi_n$ is a uniform string of length $2^n \cdot n$
- For integer function m, we will consider the function family $\{\Pi_{n,m(n)}\}$.

Subsection 2

Efficient Function Families

Efficient function families

Definition 2 (efficient function family)

An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ is efficient, if:

Samplable. $\mathcal{F}$ is samplable in polynomial-time: there exists a PPT that given 1^n, outputs (the description of) a uniform element in $\mathcal{F}_n$.

Efficient. There exists a polynomial-time algorithm that given $x \in \{0,1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

Subsection 3

Pseudorandom Functions

Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

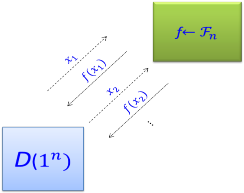
Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.



Pseudorandom Functions

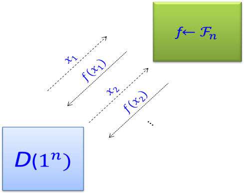
Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^{\pi}(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

$\approx_C$



Pseudorandom Functions

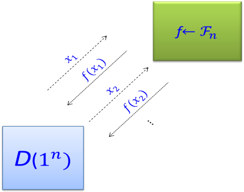
Definition 3 (pseudorandom functions (PRFs))

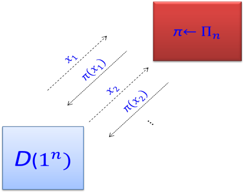
An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \mathrm{neg}(n),$$

for any oracle-aided PPT D.

$\approx_C$





Pseudorandom Functions

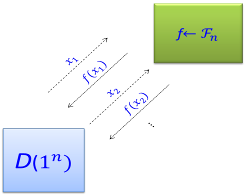
Definition 3 (pseudorandom functions (PRFs))

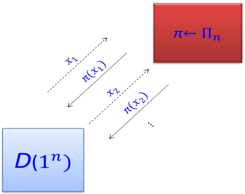
An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

$\approx_C$





- Why “oracle-aided"?

Pseudorandom Functions

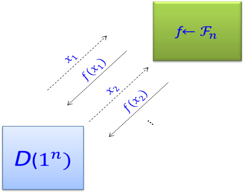
Definition 3 (pseudorandom functions (PRFs))

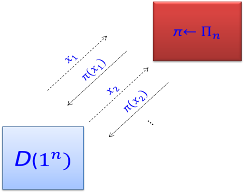
An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

$\approx_C$





- Why “oracle-aided"?
- Easy to construct (no assumption!) with logarithmic input length

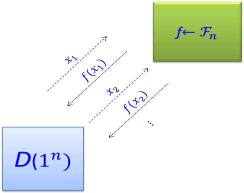
Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

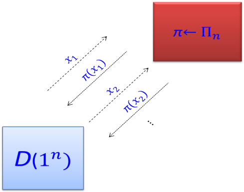
An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.



$\approx_C$



- Why “oracle-aided"?
- Easy to construct (no assumption!) with logarithmic input length
- PRFs of super logarithmic input length, which is the interesting case, imply PRGs

Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n} \left[\mathsf{D}^f(1^n) = 1 \right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}} [\mathsf{D}^\pi(1^n) = 1] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

$\approx_C$

- Why "oracle-aided"?
- Easy to construct (no assumption!) with logarithmic input length
- PRFs of super logarithmic input length, which is the interesting case, imply PRGs
- We will mainly focus on the case $m(n) = \ell(n) = n$

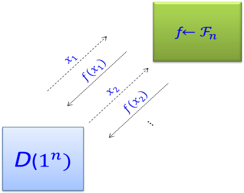
Pseudorandom Functions

Definition 3 (pseudorandom functions (PRFs))

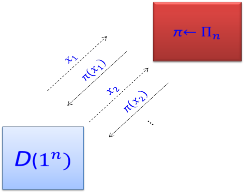
An efficient ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f(1^n) = 1\right] - \Pr_{\pi \leftarrow \Pi_{m(n),\ell(n)}}\left[\mathsf{D}^\pi(1^n) = 1\right] \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.



$\approx_C$



- Why "oracle-aided"?
- Easy to construct (no assumption!) with logarithmic input length
- PRFs of super logarithmic input length, which is the interesting case, imply PRGs
- We will mainly focus on the case $m(n) = \ell(n) = n$
- We write $\mathsf{D}^{\mathcal{F}}$ to stand for $(\mathsf{D}^f)_{f \leftarrow \mathcal{F}}$.

Section 2

PRF from OWF

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\dots,n}$
- $f_s(1) = G(s)_{n+1,\dots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Proof:

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Proof: The truth table of $f \leftarrow \mathcal{F}_n$ is $G(U_n)$, where the truth table of $\pi \leftarrow \Pi_{1,n}$ is U_{2n} $\square$

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Proof: The truth table of $f \leftarrow \mathcal{F}_n$ is $G(U_n)$, where the truth table of $\pi \leftarrow \Pi_{1,n}$ is U_{2n} $\square$

- Naturally extends to input of length $O(\log n)$:-)

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Proof: The truth table of $f \leftarrow \mathcal{F}_n$ is $G(U_n)$, where the truth table of $\pi \leftarrow \Pi_{1,n}$ is U_{2n} $\square$

- Naturally extends to input of length $O(\log n)$:-)
- Miserably fails for longer length (which is the only interesting case) :-(

Naive Construction

Let $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, and for $s \in \{0,1\}^n$ define $f_s\colon \{0,1\} \mapsto \{0,1\}^n$ by

- $f_s(0) = G(s)_{1,\ldots,n}$
- $f_s(1) = G(s)_{n+1,\ldots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathcal{F}_n = \{f_s\}_{s\in\{0,1\}^n}\}_{n\in\mathbb{N}}$ is a PRF.

Proof: The truth table of $f \leftarrow \mathcal{F}_n$ is $G(U_n)$, where the truth table of $\pi \leftarrow \Pi_{1,n}$ is U_{2n} □

- Naturally extends to input of length $O(\log n)$:-)
- Miserably fails for longer length (which is the only interesting case) :-(
- Problem, we are constructing the whole truth table, even to compute a single output

Subsection 1

The GGM Construction

The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.

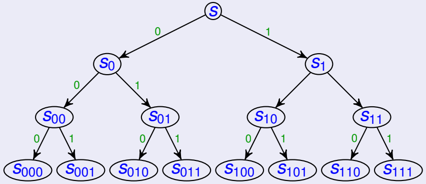
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

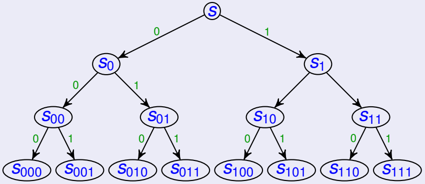
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

- Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$

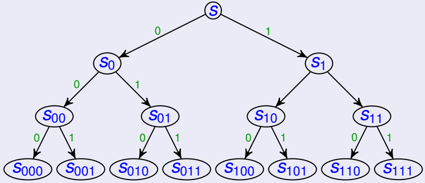
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

- Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$
- G is poly-time $\implies \mathcal{F} := \{\mathcal{F}_n = \{f_s\colon s \in \{0,1\}^n\}\}$ is efficient

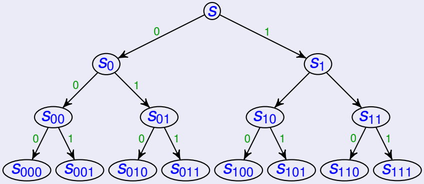
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

- Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$
- G is poly-time $\implies \mathcal{F} := \{\mathcal{F}_n = \{f_s\colon s \in \{0,1\}^n\}\}$ is efficient

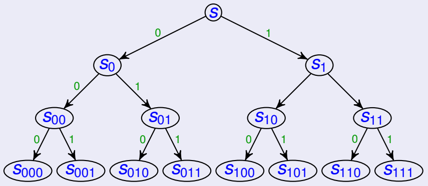
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

- Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$
- G is poly-time $\implies \mathcal{F} := \{\mathcal{F}_n = \{f_s\colon s \in \{0,1\}^n\}\}$ is efficient

Theorem 6 (Goldreich-Goldwasser-Micali (GGM))

If G is a PRG then $\mathcal{F}$ is a PRF.

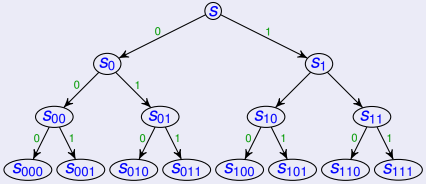
The GGM Construction

Construction 5 (GGM)

For $G\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$ and $s \in \{0,1\}^n$,

- $G_0(s) = G(s)_{1,\ldots,n}$
- $G_1(s) = G(s)_{n+1,\ldots,2n}$

For $x \in \{0,1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_{1,\ldots,k-1}))$, letting $f_s() = s$.



$s_x = f_s(x)$

- Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$
- G is poly-time $\implies \mathcal{F} := \{\mathcal{F}_n = \{f_s\colon s \in \{0,1\}^n\}\}$ is efficient

Theorem 6 (Goldreich-Goldwasser-Micali (GGM))

If G is a PRG then $\mathcal{F}$ is a PRF.

Corollary 7

OWFs imply PRFs.

Subsection 2

Proof

Proof Idea

Assume $\exists$ PPT D, $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$.

Proof Idea

Assume $\exists$ PPT D, $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$.

Fix $n \in \mathbb{N}$ and let $t = t(n)$ be a bound on the running time of $\mathsf{D}(1^n)$.

Proof Idea

Assume $\exists$ PPT D, $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$.

Fix $n \in \mathbb{N}$ and let $t = t(n)$ be a bound on the running time of $\mathsf{D}(1^n)$. We use D to construct a PPT D′ such that

$$\left|\Pr[\mathsf{D}'((U_{2n})^t) = 1] - \Pr[\mathsf{D}'(G(U_n))^t) = 1\right| > \frac{1}{np(n)},$$

where $(U_{2n})^t = U_{2n}^{(1)}, \ldots, U_{2n}^{(t)}$ and $G(U_n)^t = G(U_n^{(1)}), \ldots, G(U_n^{(t)})$.

Proof Idea

Assume $\exists$ PPT D, $p \in$ poly and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$.

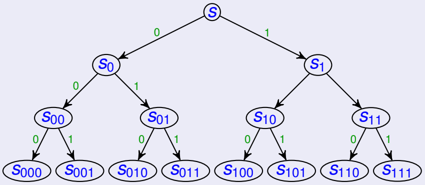
Fix $n \in \mathbb{N}$ and let $t = t(n)$ be a bound on the running time of $\mathsf{D}(1^n)$. We use D to construct a PPT D′ such that

$$\left|\Pr[\mathsf{D}'((U_{2n})^t) = 1] - \Pr[\mathsf{D}'(G(U_n))^t) = 1\right| > \frac{1}{np(n)},$$

where $(U_{2n})^t = U_{2n}^{(1)}, \ldots, U_{2n}^{(t)}$ and $G(U_n)^t = G(U_n^{(1)}), \ldots, G(U_n^{(t)})$.

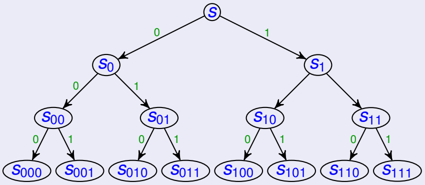
Hence, D′ violates the security of G.(?)

The Hybrid



$s_x = f_s(x)$

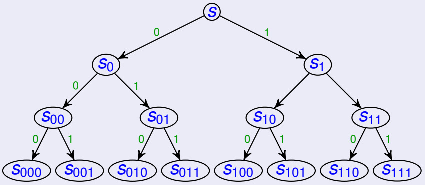
The Hybrid



$s_x = f_s(x)$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).

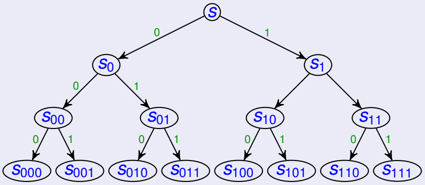
The Hybrid



$s_x = f_s(x)$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$?

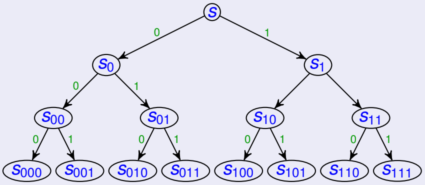
The Hybrid



$s_x = f_s(x)$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$?

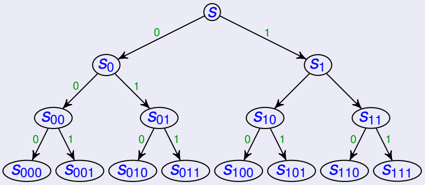
The Hybrid



$s_x = f_s(x)$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$.

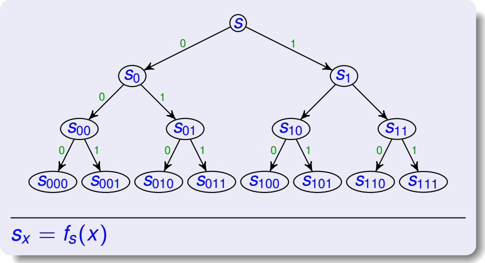
The Hybrid

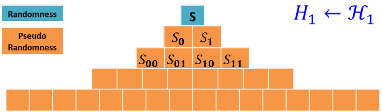


$s_x = f_s(x)$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$.

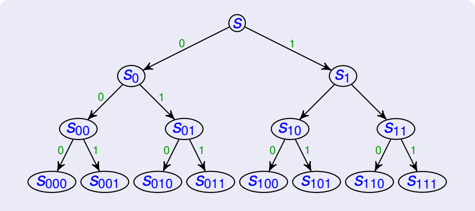
The Hybrid



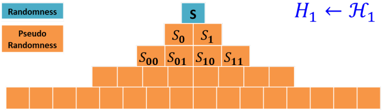


- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$. What is $\mathcal{H}_n$?

The Hybrid

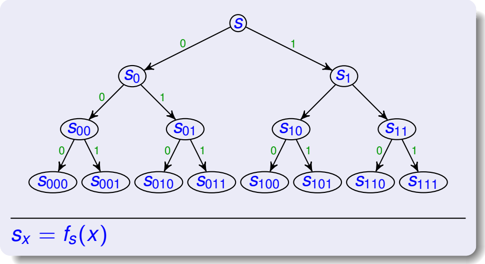


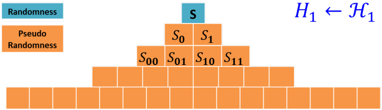
$s_x = f_s(x)$



- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$. What is $\mathcal{H}_n$? Π_n.

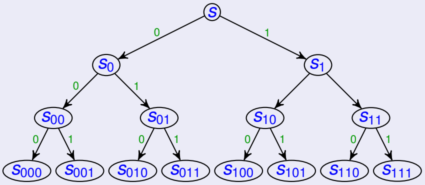
The Hybrid



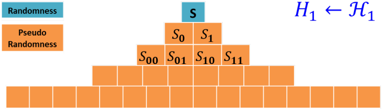


- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$. What is $\mathcal{H}_n$? Π_n.

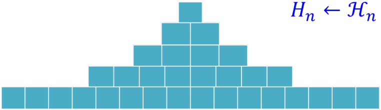
The Hybrid



$s_x = f_s(x)$

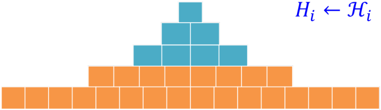


$H_1 \leftarrow \mathcal{H}_1$

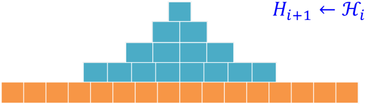


$H_n \leftarrow \mathcal{H}_n$

- $\mathcal{H}_i$: all the nodes of depth smaller than i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- What family is $\mathcal{H}_1$? $\mathcal{F}_n$. What is $\mathcal{H}_n$? Π_n.
- For some $i \in \{1, \ldots, n-1\}$, algorithm D distinguishes $\mathcal{H}_i$ from $\mathcal{H}_{i+1}$ by $\frac{1}{np(n)}$



$H_i \leftarrow \mathcal{H}_i$

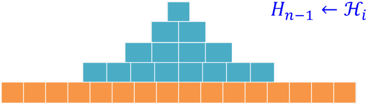


$H_{i+1} \leftarrow \mathcal{H}_{i+1}$

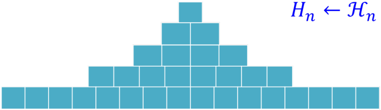
$\not\approx$

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$



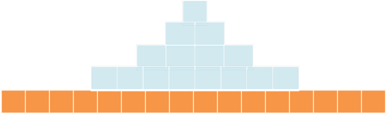
$H_{n-1} \leftarrow \mathcal{H}_{i-1}$

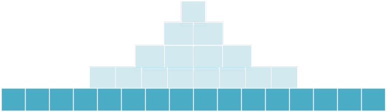


$H_n \leftarrow \mathcal{H}_n$

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$





- D distinguishes (via t samples) between
  - R – a uniform string of length $2^n \cdot n$, and
  - P - a string generated by 2^{n-1} independent calls to G

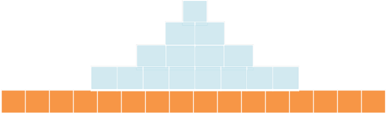
The Hybrid cont.

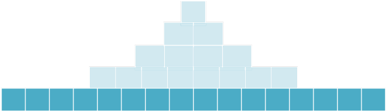
We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

- D distinguishes (via t samples) between
  - R – a uniform string of length $2^n \cdot n$, and
  - P - a string generated by 2^{n-1} independent calls to G
- We would like to use D for breaking the security of G,

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

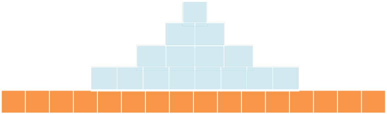


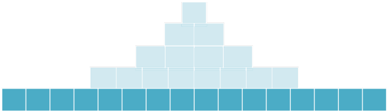


- D distinguishes (via t samples) between
  - R – a uniform string of length $2^n \cdot n$, and
  - P - a string generated by 2^{n-1} independent calls to G
- We would like to use D for breaking the security of G, but R and P seem too long :-(

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

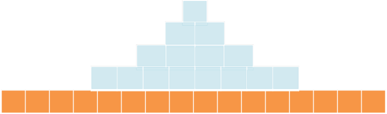


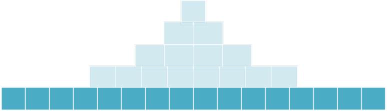


- D distinguishes (via t samples) between
  - R – a uniform string of length $2^n \cdot n$, and
  - P - a string generated by 2^{n-1} independent calls to G
- We would like to use D for breaking the security of G, but R and P seem too long :-(
- Solution: focus on the part (i.e., cells) that D *sees*

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

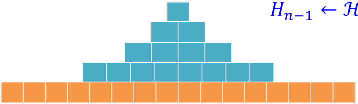




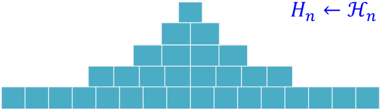
- D distinguishes (via t samples) between
  - R – a uniform string of length $2^n \cdot n$, and
  - P - a string generated by 2^{n-1} independent calls to G
- We would like to use D for breaking the security of G, but R and P seem too long :-(
- Solution: focus on the part (i.e., cells) that D *sees*

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$



$H_{n-1} \leftarrow \mathcal{H}_{i-1}$



$H_n \leftarrow \mathcal{H}_n$

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

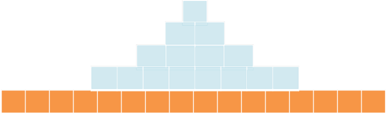
Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

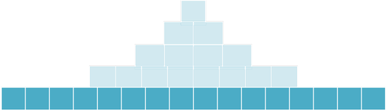
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$





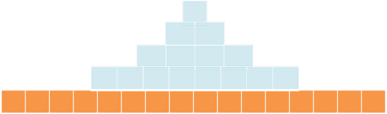
Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

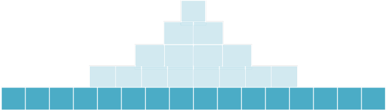
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$





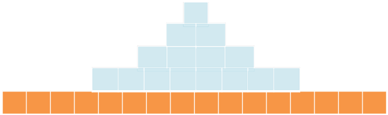
Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

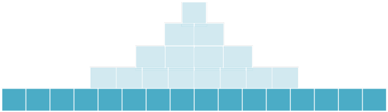
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

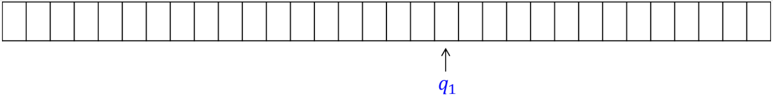




Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

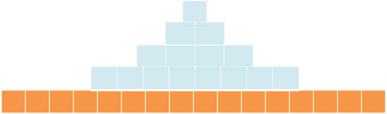
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

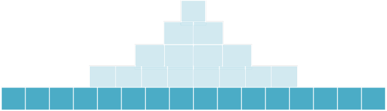
- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

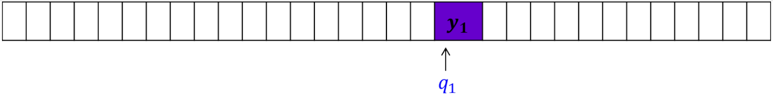




Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

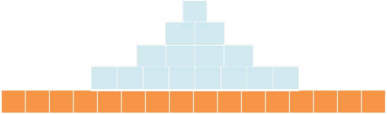
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

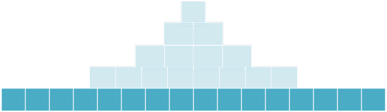
- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

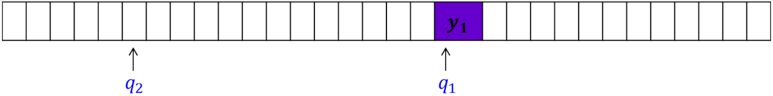




Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



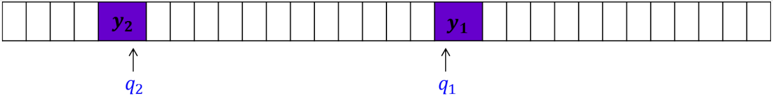
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
    - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
    - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



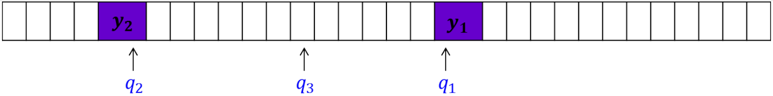
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



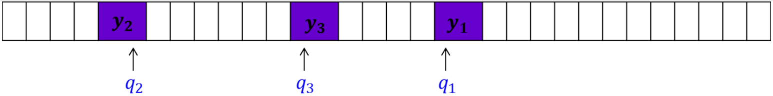
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



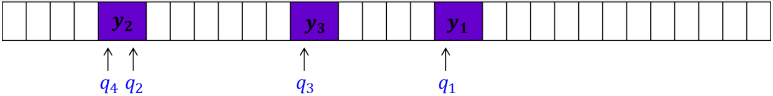
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



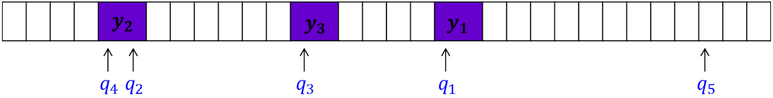
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

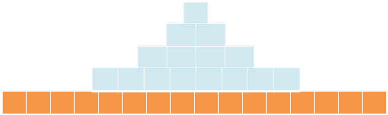
Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

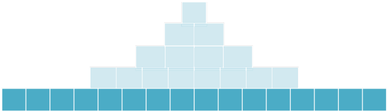
- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

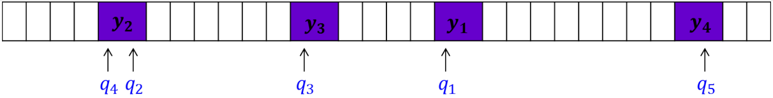




Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



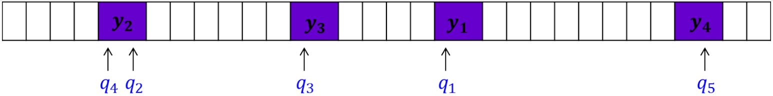
The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

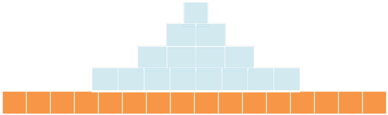
- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.

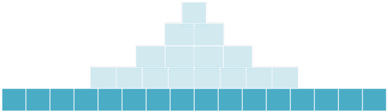


- $D'(U_{2n})^t)$ / $D'(G(U_n))^t)$ emulates D with access to R / P

The Hybrid cont.

We focus on the case where D distinguishes between $\mathcal{H}_{n-1}$ and $\mathcal{H}_n$

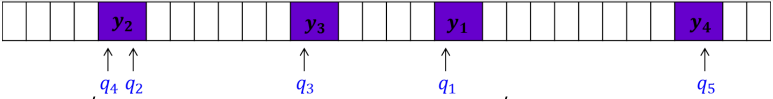




Algorithm 8 (D′ on $y_1, \ldots, y_t \in (\{0,1\}^{2n})^t$)

Emulate D. Initialize a counter $k = 0$. On the i'th query q_i made by D:

- If the cell queries by q_i'th is non-empty, answer with the content of the cell.
- Else increment k by 1 and do:
  - If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i.
  - If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i.



- $\mathsf{D}'(U_{2n})^t)$ / $\mathsf{D}'(G(U_n))^t)$ emulates D with access to R / P
- Hence, $\left|\Pr[\mathsf{D}'((U_{2n})^t) = 1] - \Pr[\mathsf{D}'(G(U_n))^t) = 1\right| > \frac{1}{np(n)}$

Part I

Pseudorandom Permutations

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{2}$$

for any oracle-aided PPT D

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{2}$$

for any oracle-aided PPT D

- Eq 2 holds for any PRF (taking the role of $\mathcal{F}$)

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{2}$$

for any oracle-aided PPT D

- Eq 2 holds for any PRF (taking the role of $\mathcal{F}$)
- Hence, PRPs are indistinguishable from PRFs...

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{2}$$

for any oracle-aided PPT D

- Eq 2 holds for any PRF (taking the role of $\mathcal{F}$)
- Hence, PRPs are indistinguishable from PRFs...
- If no one can distinguish between PRFs and PRPs, let's use PRFs

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \quad (2)$$

for any oracle-aided PPT D

- Eq 2 holds for any PRF (taking the role of $\mathcal{F}$)
- Hence, PRPs are indistinguishable from PRFs...
- If no one can distinguish between PRFs and PRPs, let's use PRFs
  - (partial) Perfect “security"

Formal Definition

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A *permutation* ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \qquad (2)$$

for any oracle-aided PPT D

- Eq 2 holds for any PRF (taking the role of $\mathcal{F}$)
- Hence, PRPs are indistinguishable from PRFs...
- If no one can distinguish between PRFs and PRPs, let's use PRFs
  - (partial) Perfect "security"
  - Inversion

Subsection 1

PRP from PRF

Feistel permutation

How does one turn a function into a permutation?

Feistel permutation

How does one turn a function into a permutation?

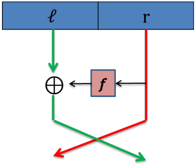
Definition 10 (LR)

For $f \colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathsf{LR}_f \colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathsf{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

Feistel permutation

How does one turn a function into a permutation?



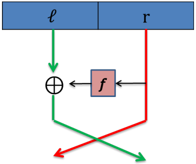
Definition 10 (LR)

For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

Feistel permutation

How does one turn a function into a permutation?



Definition 10 (LR)

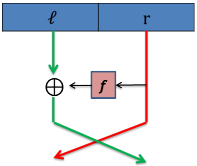
For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- LR_f is a permutation: $\mathrm{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$

Feistel permutation

How does one turn a function into a permutation?



Definition 10 (LR)

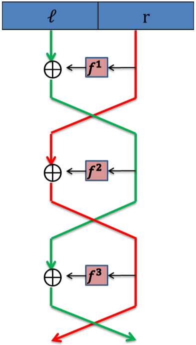
For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- LR_f is a permutation: $\mathrm{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$
- LR_f is efficiently computable and invertible given oracle access to f

Feistel permutation

How does one turn a function into a permutation?



Definition 10 (LR)

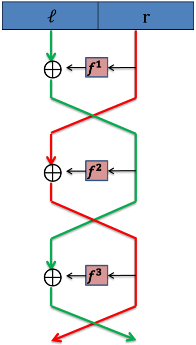
For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- LR_f is a permutation: $\mathrm{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$
- LR_f is efficiently computable and invertible given oracle access to f

Feistel permutation

How does one turn a function into a permutation?



Definition 10 (LR)

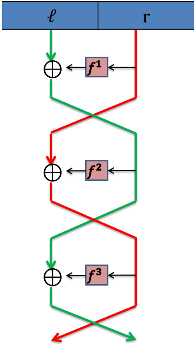
For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- LR_f is a permutation: $\mathrm{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$
- LR_f is efficiently computable and invertible given oracle access to f
- For $i \in \mathbb{N}$ and $f^1, \ldots, f^i$, define $\mathrm{LR}_{f^1,\ldots,f^i}\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ by

Feistel permutation

How does one turn a function into a permutation?



Definition 10 (LR)

For $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, let $\mathrm{LR}_f\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be defined by

$$\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- LR_f is a permutation: $\mathrm{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$
- LR_f is efficiently computable and invertible given oracle access to f
- For $i \in \mathbb{N}$ and $f^1, \ldots, f^i$, define $\mathrm{LR}_{f^1,\ldots,f^i}\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ by

  $\mathrm{LR}_{f^1,\ldots,f^i}(\ell, r) = (r^{i-1}, f^i(r^{i-1}) \oplus \ell^{i-1})$, for $(\ell^{i-1}, r^{i-1}) = \mathrm{LR}_{f^1,\ldots,f^{i-1}}(\ell, r)$.
  (letting $(\ell^0, r^0) = (\ell, r)$)

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathrm{LR}^1_{\mathcal{F}}$ pseudorandom?

Luby-Rackoff Thm.

Recall $\mathsf{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathsf{LR}^i(\mathcal{F}) = \{\mathsf{LR}^i_{\mathcal{F}_n} = \{\mathsf{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathsf{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathsf{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathsf{LR}^2_{\mathcal{F}}$?

Luby-Rackoff Thm.

Recall $\mathsf{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathsf{LR}^i(\mathcal{F}) = \{\mathsf{LR}^i_{\mathcal{F}_n} = \{\mathsf{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathsf{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathsf{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathsf{LR}^2_{\mathcal{F}}$?

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathrm{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathrm{LR}^2_{\mathcal{F}}$? $\mathrm{LR}_{f^1,f^2}(0^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
  and $\mathrm{LR}_{f^1,f^2}(1^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let
$\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathrm{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathrm{LR}^2_{\mathcal{F}}$? $\mathrm{LR}_{f^1,f^2}(0^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
  and $\mathrm{LR}_{f^1,f^2}(1^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$
- $\mathrm{LR}^3_{\mathcal{F}}$?

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let
$\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathrm{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathrm{LR}^2_{\mathcal{F}}$? $\mathrm{LR}_{f^1,f^2}(0^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
  and $\mathrm{LR}_{f^1,f^2}(1^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$
- $\mathrm{LR}^3_{\mathcal{F}}$?

Luby-Rackoff Thm.

Recall $\mathsf{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathsf{LR}^i(\mathcal{F}) = \{\mathsf{LR}^i_{\mathcal{F}_n} = \{\mathsf{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathsf{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathsf{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathsf{LR}^2_{\mathcal{F}}$? $\mathsf{LR}_{f^1,f^2}(0^n, 0^n) = \mathsf{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
  and $\mathsf{LR}_{f^1,f^2}(1^n, 0^n) = \mathsf{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$
- $\mathsf{LR}^3_{\mathcal{F}}$?

Theorem 12 (Luby-Rackoff)

Assuming that $\mathcal{F}$ is a PRF, then $\mathsf{LR}^3_{\mathcal{F}}$ is a PRP

Luby-Rackoff Thm.

Recall $\mathrm{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i_{\mathcal{F}_n} = \{\mathrm{LR}_{f^1,\ldots,f^i} \colon f^1, \ldots, f^i \in \mathcal{F}_n\}\}$,

- $\mathrm{LR}^i_{\mathcal{F}}$ is always a permutation family, and is efficient if $\mathcal{F}$ is.
- Is $\mathrm{LR}^1_{\mathcal{F}}$ pseudorandom?
- $\mathrm{LR}^2_{\mathcal{F}}$? $\mathrm{LR}_{f^1,f^2}(0^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
  and $\mathrm{LR}_{f^1,f^2}(1^n, 0^n) = \mathrm{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$
- $\mathrm{LR}^3_{\mathcal{F}}$?

Theorem 12 (Luby-Rackoff)

Assuming that $\mathcal{F}$ is a PRF, then $\mathrm{LR}^3_{\mathcal{F}}$ is a PRP

- $\mathrm{LR}^4(\mathcal{F})$ is pseudorandom even if inversion queries are allowed

Proving Luby-Rackoff

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?

Proving Luby-Rackoff

It suffices to prove that $\mathsf{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathsf{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$?

Proving Luby-Rackoff

It suffices to prove that $\mathsf{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathsf{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$?

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathrm{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathrm{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is

Proving Luby-Rackoff

It suffices to prove that $\mathsf{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathsf{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathsf{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathrm{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathrm{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q-query D,

$$|\Pr[\mathrm{D}^{\mathrm{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathrm{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \in O(q^2/2^n).$$

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathrm{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathrm{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q-query D,

$$|\Pr[\mathrm{D}^{\mathrm{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathrm{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \in O(q^2/2^n).$$

- We assume for simplicity that D is *deterministic*, *non-repeating* and *non-adaptive*.

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathrm{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathrm{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q-query D,

$$|\Pr[\mathrm{D}^{\mathrm{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathrm{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \in O(q^2/2^n).$$

- We assume for simplicity that D is *deterministic*, *non-repeating* and *non-adaptive*.
- Let $x_1, \ldots, x_q$ be D's queries.

Proving Luby-Rackoff

It suffices to prove that $\mathrm{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathrm{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathrm{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q-query D,

$$|\Pr[\mathrm{D}^{\mathrm{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathrm{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \in O(q^2/2^n).$$

- We assume for simplicity that D is *deterministic*, *non-repeating* and *non-adaptive*.
- Let $x_1, \ldots, x_q$ be D's queries.
- We show $(f(x_1), \ldots, f(x_q))_{f \leftarrow \mathrm{LR}^3(\Pi_n)}$ is $O(q^2/2^n)$ close (i.e., in statistical distance) to $(f(x_1), \ldots, f(x_q))_{f \leftarrow \widetilde{\Pi}}$

Proving Luby-Rackoff

It suffices to prove that $\mathsf{LR}^3_{\Pi_n}$ is pseudorandom (?)

- How would you prove that?
- Maybe $\mathsf{LR}^3(\Pi_n) \equiv \widetilde{\Pi}_{2n}$? description length of element in $\mathsf{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\widetilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left((\frac{2^{2n}}{e})^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q-query D,

$$|\Pr[\mathsf{D}^{\mathsf{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \in O(q^2/2^n).$$

- We assume for simplicity that D is *deterministic*, *non-repeating* and *non-adaptive*.
- Let $x_1, \ldots, x_q$ be D's queries.
- We show $(f(x_1), \ldots, f(x_q))_{f \leftarrow \mathsf{LR}^3(\Pi_n)}$ is $O(q^2/2^n)$ close (i.e., in statistical distance) to $(f(x_1), \ldots, f(x_q))_{f \leftarrow \widetilde{\Pi}}$
- To do that, we show both distributions are $O(q^2/2^n)$ close to $\mathit{Distinct} := \left((z_1, \ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0\right)$.

Reminder: Statistical Distance

Definition 14

The statistical distance between distributions P and Q over $\mathcal{U}$, is defined by

$$\mathsf{SD}(P,Q) = \frac{1}{2} \cdot \sum_{u \in \mathcal{U}} |P(u) - Q(u)| = \max_{\mathcal{S} \subseteq \mathcal{U}} \{ \Pr_Q[\mathcal{S}] - \Pr_P[\mathcal{S}] \}$$

Reminder: Statistical Distance

Definition 14

The statistical distance between distributions P and Q over $\mathcal{U}$, is defined by

$$\mathrm{SD}(P,Q) = \frac{1}{2} \cdot \sum_{u \in \mathcal{U}} |P(u) - Q(u)| = \max_{\mathcal{S} \subseteq \mathcal{U}} \{ \Pr_{Q}[\mathcal{S}] - \Pr_{P}[\mathcal{S}] \}$$

In case $\mathrm{SD}(P,Q) \leq \varepsilon$, we say that P and Q are ε close.

Reminder: Statistical Distance

Definition 14

The statistical distance between distributions P and Q over $\mathcal{U}$, is defined by

$$\mathsf{SD}(P,Q) = \frac{1}{2} \cdot \sum_{u \in \mathcal{U}} |P(u) - Q(u)| = \max_{\mathcal{S} \subseteq \mathcal{U}} \{\Pr_Q[\mathcal{S}] - \Pr_P[\mathcal{S}]\}$$

In case $\mathsf{SD}(P,Q) \leq \varepsilon$, we say that P and Q are ε close.

Fact 15

Let $\mathcal{E}$ be an event (i.e., set) and assume $\mathsf{SD}(P|_{\neg\mathcal{E}}, Q) \leq \delta_1$ and $\Pr_P[\mathcal{E}] \leq \delta_2$. Then $\mathsf{SD}(P,Q) \leq \delta_1 + \delta_2$

Proving Fact 15

Proving Fact 15

For any set $\mathcal{S}$, it holds that

$$\Pr_{P}[\mathcal{S}] = \Pr_{P}[\mathcal{E}] \cdot \Pr_{P|_{\mathcal{E}}}[\mathcal{S}] + \Pr_{P}[\neg\mathcal{E}] \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \geq (1-\delta_2) \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \tag{3}$$

Proving Fact 15

For any set $\mathcal{S}$, it holds that

$$\Pr_P[\mathcal{S}] = \Pr_P[\mathcal{E}] \cdot \Pr_{P|_{\mathcal{E}}}[\mathcal{S}] + \Pr_P[\neg\mathcal{E}] \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \tag{3}$$
$$\geq (1 - \delta_2) \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}]$$

Hence,

$$\Pr_Q[\mathcal{S}] - \Pr_P[\mathcal{S}] \leq \Pr_Q[\mathcal{S}] - (1 - \delta_2) \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \tag{4}$$
$$\leq \Pr_Q[\mathcal{S}] - \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] + \delta_2$$

Proving Fact 15

For any set $\mathcal{S}$, it holds that

$$\Pr_P[\mathcal{S}] = \Pr_P[\mathcal{E}] \cdot \Pr_{P|_{\mathcal{E}}}[\mathcal{S}] + \Pr_P[\neg\mathcal{E}] \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \geq (1-\delta_2) \cdot \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \tag{3}$$

Hence,

$$\Pr_Q[\mathcal{S}] - \Pr_P[\mathcal{S}] \leq \Pr_Q[\mathcal{S}] - (1-\delta_2)\Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] \leq \Pr_Q[\mathcal{S}] - \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}] + \delta_2 \tag{4}$$

Thus,

$$\mathsf{SD}(P,Q) = \max_{\mathcal{S}}\{\Pr_Q[\mathcal{S}] - \Pr_P[\mathcal{S}]\} \leq \max_{\mathcal{S}}\{\Pr_Q[\mathcal{S}] - \Pr_{P|_{\neg\mathcal{E}}}[\mathcal{S}]\} + \delta_2 = \delta_1 + \delta_2.$$

$(f(x_0), \ldots, f(x_q))_{f \leftarrow \widetilde{\Pi}}$ is close to *Distinct*

$(f(x_0),\ldots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is close to *Distinct*

Recall $\mathit{Distinct} := \big((z_1,\ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0\big)$.

$(f(x_0), \ldots, f(x_q))_{f \leftarrow \widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= \big((z_1, \ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0\big)$.

For $f \in \widetilde{\Pi}$, let $Bad(f) := \exists i \neq j \colon f(x_i)_0 = f(x_j)_0$.

$(f(x_0),\ldots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= \left((z_1,\ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j\colon (z_i)_0 \neq (z_j)_0\right)$.

For $f \in \widetilde{\Pi}$, let $Bad(f) := \exists i \neq j\colon f(x_i)_0 = f(x_j)_0$.

Claim 16

$\Pr_{f\leftarrow\widetilde{\Pi}}[Bad(f)] \leq \frac{\binom{q}{2}}{2^n} \leq \frac{q^2}{2^n}$

$(f(x_0), \ldots, f(x_q))_{f \leftarrow \widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= \big((z_1, \ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0\big)$.

For $f \in \widetilde{\Pi}$, let $Bad(f) := \exists i \neq j \colon f(x_i)_0 = f(x_j)_0$.

Claim 16

$\Pr_{f \leftarrow \widetilde{\Pi}}[Bad(f)] \leq \frac{\binom{q}{2}}{2^n} \leq \frac{q^2}{2^n}$

Proof: ?

$(f(x_0),\ldots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= \left((z_1,\ldots z_q)\leftarrow(\{0,1\}^{2n})^q \mid \forall i\neq j\colon (z_i)_0 \neq (z_j)_0\right)$.

For $f\in\widetilde{\Pi}$, let $Bad(f) := \exists i\neq j\colon f(x_i)_0 = f(x_j)_0$.

Claim 16

$\Pr_{f\leftarrow\widetilde{\Pi}}[Bad(f)] \leq \frac{\binom{q}{2}}{2^n} \leq \frac{q^2}{2^n}$

Proof: ?

Claim 17

$\left((f(x_0),\ldots,f(x_q)); f\leftarrow\widetilde{\Pi} \mid \neg\,\mathrm{Bad}(f)\right) \equiv Distinct$

$(f(x_0),\ldots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= ((z_1,\ldots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0)$.

For $f \in \widetilde{\Pi}$, let $Bad(f) := \exists i \neq j \colon f(x_i)_0 = f(x_j)_0$.

Claim 16

$\Pr_{f\leftarrow\widetilde{\Pi}}[Bad(f)] \leq \frac{\binom{q}{2}}{2^n} \leq \frac{q^2}{2^n}$

Proof: ?

Claim 17

$\left((f(x_0),\ldots,f(x_q)); f \leftarrow \widetilde{\Pi} \mid \neg \mathrm{Bad}(f)\right) \equiv \mathit{Distinct}$

Proof: ?

$(f(x_0),\dots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is close to *Distinct*

Recall *Distinct* $:= \left((z_1,\dots z_q) \leftarrow (\{0,1\}^{2n})^q \mid \forall i \neq j \colon (z_i)_0 \neq (z_j)_0\right)$.

For $f \in \widetilde{\Pi}$, let $Bad(f) := \exists i \neq j \colon f(x_i)_0 = f(x_j)_0$.

Claim 16

$\Pr_{f\leftarrow\widetilde{\Pi}}[Bad(f)] \leq \frac{\binom{q}{2}}{2^n} \leq \frac{q^2}{2^n}$

Proof: ?

Claim 17

$\left((f(x_0),\dots,f(x_q)); f \leftarrow \widetilde{\Pi} \mid \neg\,\mathrm{Bad}(f)\right) \equiv$ *Distinct*

Proof: ?

By Fact 15, $(f(x_0),\dots,f(x_q))_{f\leftarrow\widetilde{\Pi}}$ is $\frac{q^2}{2^n}$ close to *Distinct*

$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

$(f(x_0), \ldots, f(x_q))_{f \leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0), \ldots, (\ell_q^0, r_q^0) = (x_1, \ldots, x_k)$.

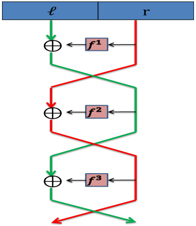
$(f(x_0), \ldots, f(x_q))_{f \leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0), \ldots, (\ell_q^0, r_q^0) = (x_1, \ldots, x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



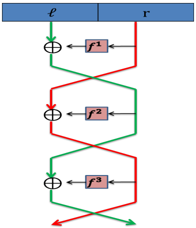
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Claim 18

$\Pr_{f^1\leftarrow \Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j\colon r_i^1 = r_j^1\right] \le \frac{\binom{q}{2}}{2^n}$

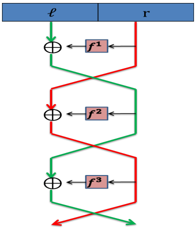
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof:

Claim 18

$\Pr_{f^1\leftarrow\Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j \colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$

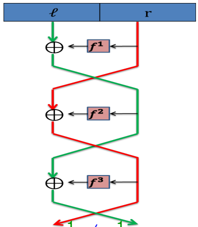
$(f(x_0), \ldots, f(x_q))_{f \leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0), \ldots, (\ell_q^0, r_q^0) = (x_1, \ldots, x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$

Claim 18

$$\Pr_{f^1 \leftarrow \Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j \colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

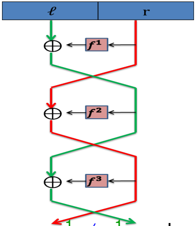
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Claim 18

$\Pr_{f^1\leftarrow\Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j \colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$

Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and

$r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ □

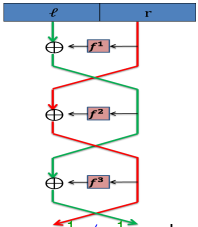
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and $r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ $\square$

Claim 18

$$\Pr_{f^1\leftarrow\Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j\colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

Claim 19

$$\Pr_{(f^1,f^2)\leftarrow\Pi_n^2}\left[\mathrm{Bad}^2 := \exists i \neq j\colon r_i^1 = r_j^1 \vee r_i^2 = r_j^2\right] \leq 2\cdot\frac{\binom{q}{2}}{2^n} \in O\left(\frac{q^2}{2^n}\right)$$

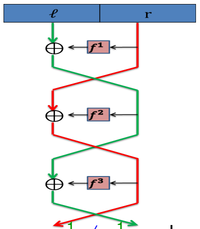
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and
$r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ $\square$

Claim 18

$$\Pr_{f^1\leftarrow \Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j \colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

Claim 19

$$\Pr_{(f^1,f^2)\leftarrow \Pi_n^2}\left[\mathrm{Bad}^2 := \exists i \neq j \colon r_i^1 = r_j^1 \vee r_i^2 = r_j^2\right] \leq 2\cdot \frac{\binom{q}{2}}{2^n} \in O(\tfrac{q^2}{2^n})$$

Proof:

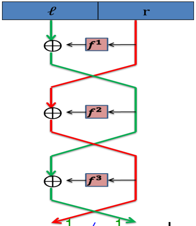
$(f(x_0), \ldots, f(x_q))_{f \leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0), \ldots, (\ell_q^0, r_q^0) = (x_1, \ldots, x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and

$r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ $\square$

Claim 18

$$\Pr_{f^1 \leftarrow \Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j \colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

Claim 19

$$\Pr_{(f^1, f^2) \leftarrow \Pi_n^2}\left[\mathrm{Bad}^2 := \exists i \neq j \colon r_i^1 = r_j^1 \vee r_i^2 = r_j^2\right] \leq 2 \cdot \frac{\binom{q}{2}}{2^n} \in O\left(\frac{q^2}{2^n}\right)$$

Proof: similar to the above

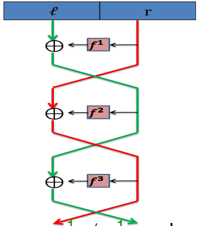
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1,f^2,f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and

$r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ $\square$

Claim 18

$$\Pr_{f^1\leftarrow\Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j\colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

Claim 19

$$\Pr_{(f^1,f^2)\leftarrow\Pi_n^2}\left[\mathrm{Bad}^2 := \exists i \neq j\colon r_i^1 = r_j^1 \vee r_i^2 = r_j^2\right] \leq 2\cdot\frac{\binom{q}{2}}{2^n} \in O\left(\frac{q^2}{2^n}\right)$$

Proof: similar to the above

Claim 20

$$\left((\ell_1^3, r_1^3),\ldots,(\ell_q^3, r_q^3) \mid \neg\,\mathrm{Bad}^2\right) \equiv \textit{Distinct}$$

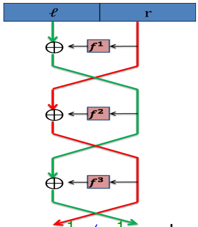
$(f(x_0),\ldots,f(x_q))_{f\leftarrow \mathrm{LR}^3(\Pi_n)}$ is close to *Distinct*

Let $(\ell_1^0, r_1^0),\ldots,(\ell_q^0, r_q^0) = (x_1,\ldots,x_k)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	$\ldots$	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	$\ldots$	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^0	$\ldots$	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^0	$\ldots$	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.



Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1$ and

$r_i^0 \neq r_j^0 \implies \Pr_{f^1}\left[r_i^1 = r_j^1\right] = 2^{-n}$ □

Claim 18

$$\Pr_{f^1\leftarrow\Pi_n}\left[\mathrm{Bad}^1 := \exists i \neq j\colon r_i^1 = r_j^1\right] \leq \frac{\binom{q}{2}}{2^n}$$

Proof: similar to the above

Claim 19

$$\Pr_{(f^1,f^2)\leftarrow\Pi_n^2}\left[\mathrm{Bad}^2 := \exists i \neq j\colon r_i^1 = r_j^1 \vee r_i^2 = r_j^2\right] \leq 2\cdot\frac{\binom{q}{2}}{2^n} \in O\left(\frac{q^2}{2^n}\right)$$

Proof: ?

Claim 20

$$\left((\ell_1^3, r_1^3),\ldots,(\ell_q^3, r_q^3) \mid \neg\,\mathrm{Bad}^2\right) \equiv \textit{Distinct}$$

Proving Claim 20

Proving Claim 20

Let $\mathcal{S} = \{(z_1, \ldots, z_q) \in (\{0,1\}^n)^q \colon \forall i \neq j \colon z_i \neq z_j\}$.

Proving Claim 20

Let $\mathcal{S} = \{(z_1, \ldots, z_q) \in (\{0,1\}^n)^q \colon \forall i \neq j \colon z_i \neq z_j\}$.

Claim 21

$\left((\ell_1^3, \ldots, \ell_q^3) \mid \neg\,\mathrm{Bad}^2\right)$ is uniform over $\mathcal{S}$.

Proving Claim 20

Let $\mathcal{S} = \{(z_1, \ldots, z_q) \in (\{0,1\}^n)^q \colon \forall i \neq j \colon z_i \neq z_j\}$.

Claim 21

$\left((\ell_1^3, \ldots, \ell_q^3) \mid \neg \mathrm{Bad}^2\right)$ is uniform over $\mathcal{S}$.

Proof:

Proving Claim 20

Let $\mathcal{S} = \{(z_1, \ldots, z_q) \in (\{0,1\}^n)^q \colon \forall i \neq j \colon z_i \neq z_j\}$.

Claim 21

$\left((\ell_1^3, \ldots, \ell_q^3) \mid \neg \operatorname{Bad}^2\right)$ is uniform over $\mathcal{S}$.

Proof: For any $\mathbf{z} = (z_1, \ldots, z_q) \in (\{0,1\}^n)^q$ and $\pi \in \Pi_n$:

$\Pr\left[(\ell_1^3, \ldots, \ell_q^3) = \mathbf{z}\right] = \Pr\left[(\ell_1^3, \ldots, \ell_q^3) = \pi(\mathbf{z}) := (\pi(z_1), \ldots, \pi(z_q))\right]$ □

Section 3

Applications

General paradigm

Design a scheme assuming that you have random functions, and the realize them using PRFs.

Subsection 1

Private-key Encryption

Private-key Encryption

Construction 22 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D}))$:

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

Private-key Encryption

Construction 22 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D})$):

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?

Construction 22 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D}))$:

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?
- Proof of security?

Conclusion

- We constructed PRFs and PRPs from length-doubling PRG (and thus from one-way functions)

Conclusion

- We constructed PRFs and PRPs from length-doubling PRG (and thus from one-way functions)
- Main question: find a simpler, more efficient construction

Conclusion

- We constructed PRFs and PRPs from length-doubling PRG (and thus from one-way functions)
- Main question: find a simpler, more efficient construction

  or at least, a less adaptive one