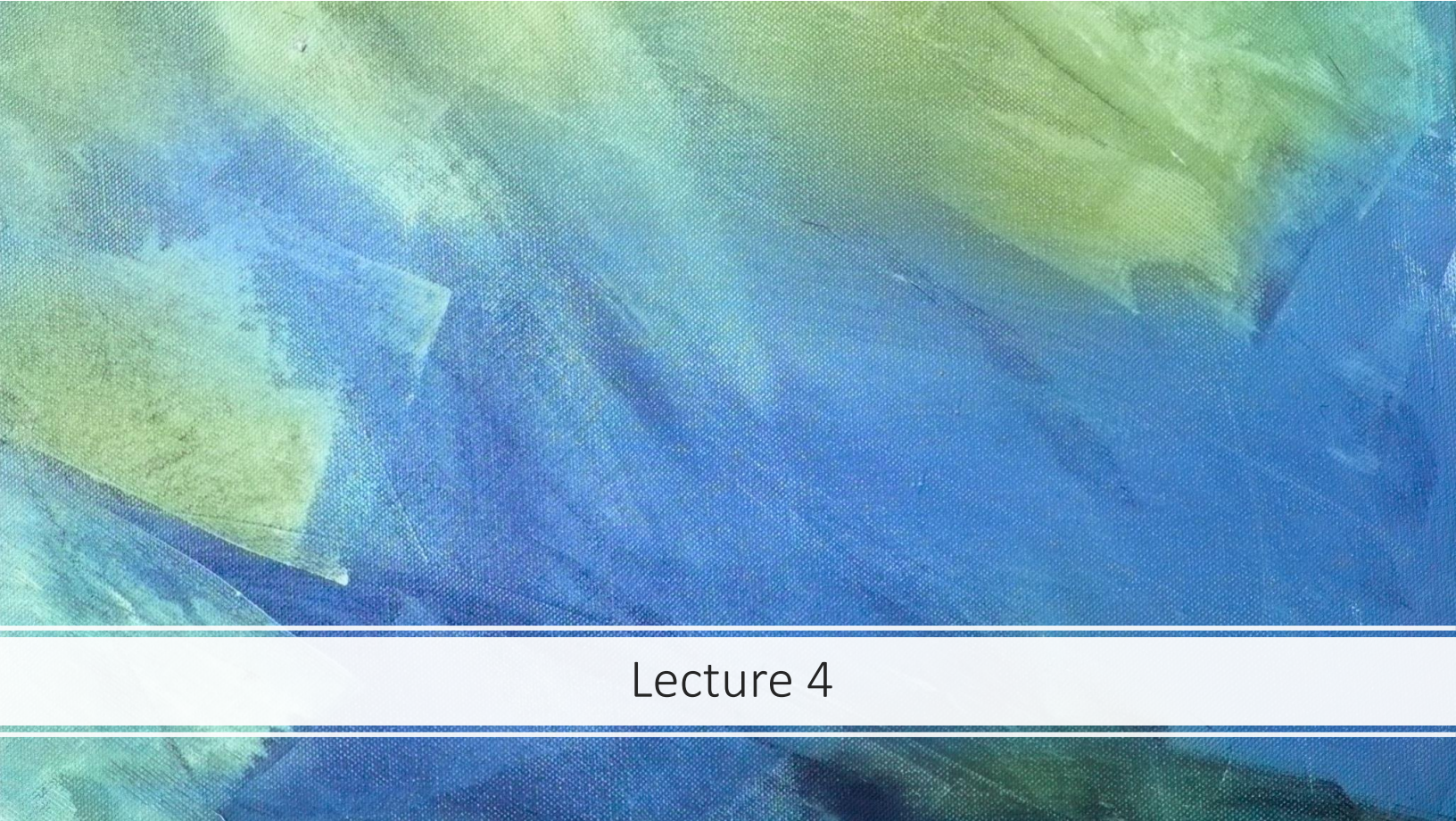
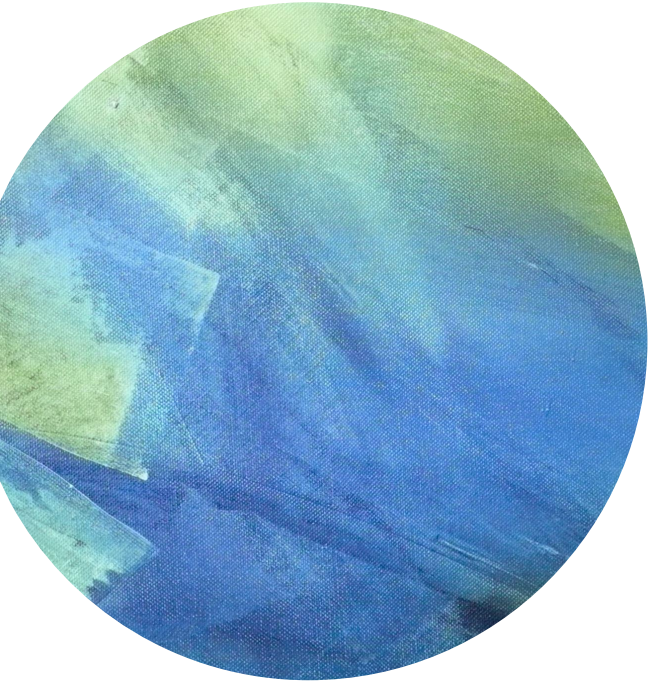


Lecture 4



Introduction

Pseudorandom

Stream Ciphers

Block Ciphers:
PRFs, PRPs

Administrative Details

- Scribe volunteer

Pseudorandom Generators

PRGs are "parameterized" by security parameter λ which represents key length (input)

- **PRG** becomes "more secure" as λ increases

$$Pr[\text{success}] = \frac{1}{2^{\lambda}}$$

Seed lengths and output lengths grow with λ

For every $\lambda=1,2,3,\ldots$ there is a different PRG G_λ:

$$G_\lambda : K_\lambda \longrightarrow \{0,1\}^{n(\lambda)}$$

(in the lectures we will often omit λ)

n is a fixed polynomial,

$n = 3\lambda$

$n > \lambda$

Rigorous Definitions

A PRG $= \{G_\lambda\}_{\lambda \in \mathbb{N}}$ is secure iff

for every poly-sized adversary (PPT) $\mathcal{A}$,

there exists a function $\varepsilon = \varepsilon(\lambda)$ s.t.

① $\left| \Pr_{k \leftarrow K}[\mathcal{A}(G_\lambda(k)) = 1] - \Pr_{r \leftarrow \{0,1\}^n}[\mathcal{A}(r) = 1] \right| \leq \varepsilon(\lambda)$

② ε should be a negligible function. $\left(\exists \lambda_0 \text{ s.t. } \forall \lambda \geq \lambda_0 \; \varepsilon(\lambda) < \frac{1}{\lambda^c} \; \forall c\right)$

$\frac{1}{\lambda^{\log \lambda}}$

Rigorous Definitions

Let P_1 and P_2 be two distributions over $\{0,1\}^n$

Def: We say that P_1 and P_2 are

computationally indistinguishable (denoted $\approx_c$)

$\forall$ PPT $\mathcal{A}$

$$\left| \Pr_{s \leftarrow P_1}[\mathcal{A}(s) = 1] - \Pr_{s \leftarrow P_2}[\mathcal{A}(s) = 1] \right| = \text{negl}$$

Example: a PRG is secure if $\{ k \leftarrow \{0,1\}^{\lambda} : G(k) \} \approx_c \text{uniform}(\{0,1\}^{n(\lambda)})$

$\leq 2^{\lambda}$ $\quad$ $2^{n(\lambda)}$

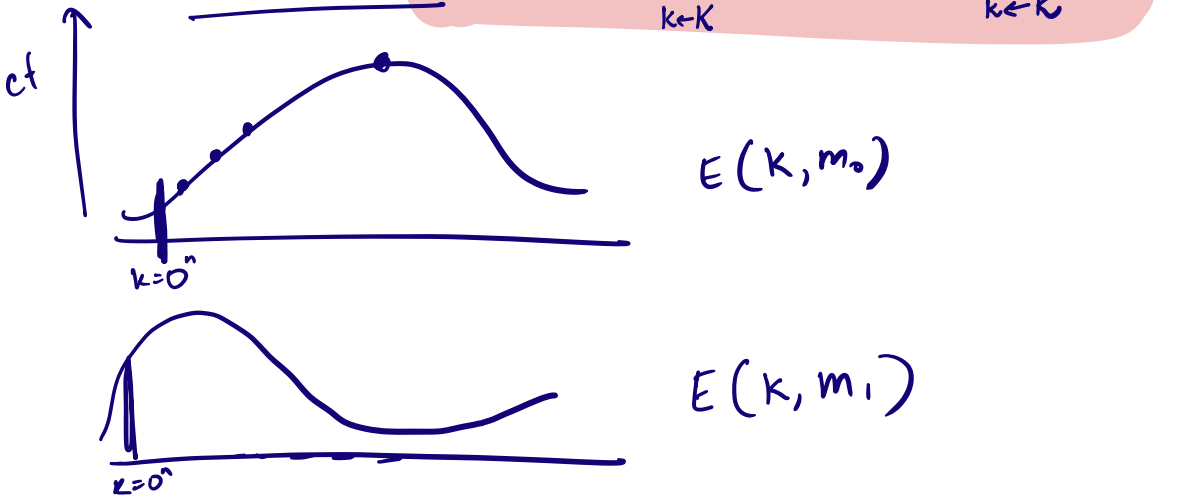
"stretch": $(n(\lambda) - \lambda)$

Semantic Security

for fixed k, $Enc(k, m_0) \neq Enc(k, m_1)$ o/w cannot decrypt!

Def: a cipher is **semantically secure** if

for all $m_0, m_1 \in M$: $\{ E(k,m_0) \}_{k \leftarrow K} \approx_c \{ E(k,m_1) \}_{k \leftarrow K}$



Semantic Security

THM.

PRG-based stream cipher is semantically secure.

$Enc(k,m) = m \oplus G(k)$

$Dec(k,c) = c \oplus G(k)$

i.e. for all $m_0, m_1 \in M$: $\{Enc(k,m_0)\} \approx_c \{Enc(k,m_1)\}$

Claim :

$(G(k) \oplus m_0) \approx_c (G(k) \oplus m_1)$ by PRG

Sample $r \xleftarrow{\$} \{0,1\}^n$: $\{r \oplus m_0\}_{r \xleftarrow{\$} \{0,1\}^n} = \{r \oplus m_1\}_{r \xleftarrow{\$} \{0,1\}^n}$.

unif. distribution over $\{0,1\}^n$

uniform dist. over $\{0,1\}^n$

Semantic Security

THM.

PRG-based stream cipher is semantically secure.

$Enc(k,m) = m \oplus G(k)$

$Dec(k,c) = c \oplus G(k)$

i.e. for all $m_0, m_1 \in M$: $\{Enc(k,m_0)\} \approx_c \{Enc(k,m_1)\}$

Claim: $\mathcal{D}_1(G(k) \oplus m_0) \approx_c \mathcal{D}_2(\{r \oplus m_0\}_{r \xleftarrow{R} \{0,1\}^n})$

Sample $r \xleftarrow{R} \{0,1\}^n$

↓

unif. distribution over $\{0,1\}^n$

Suppose $\mathcal{D}_1 \not\approx_c \mathcal{D}_2$.

$\exists \mathcal{A}$, poly s.t.

$\left| Pr[\mathcal{A}(G(k) \oplus m_0) = 1] - Pr[\mathcal{A}(r \oplus m_0) = 1] \right| \geq \frac{1}{poly}$

build B s.t.

$Pr[B(G(k)) = 1] - Pr[B(r) = 1] \geq \frac{1}{poly}$

Semantic Security

THM.

PRG-based stream cipher is semantically secure.

$Enc(k,m) = m \oplus G(k)$

$Dec(k,c) = c \oplus G(k)$

i.e. for all $m_0, m_1 \in M$: $\{Enc(k,m_0)\} \approx_c \{Enc(k,m_1)\}$

Claim: $\mathcal{D}_1 \left(G(k) \oplus m_0 \right) \approx_c \mathcal{D}_2 \left(\{ r \oplus m_0 \}_{r \xleftarrow{\$} \{0,1\}^n} \right)$

Sample $r \xleftarrow{\$} \{0,1\}^n$

$\downarrow$

unif. distribution over $\{0,1\}^n$

B (y)

Run $\mathcal{A}(y \oplus m)$

O/p the same bit as $\mathcal{A}$

B breaks PRG security.

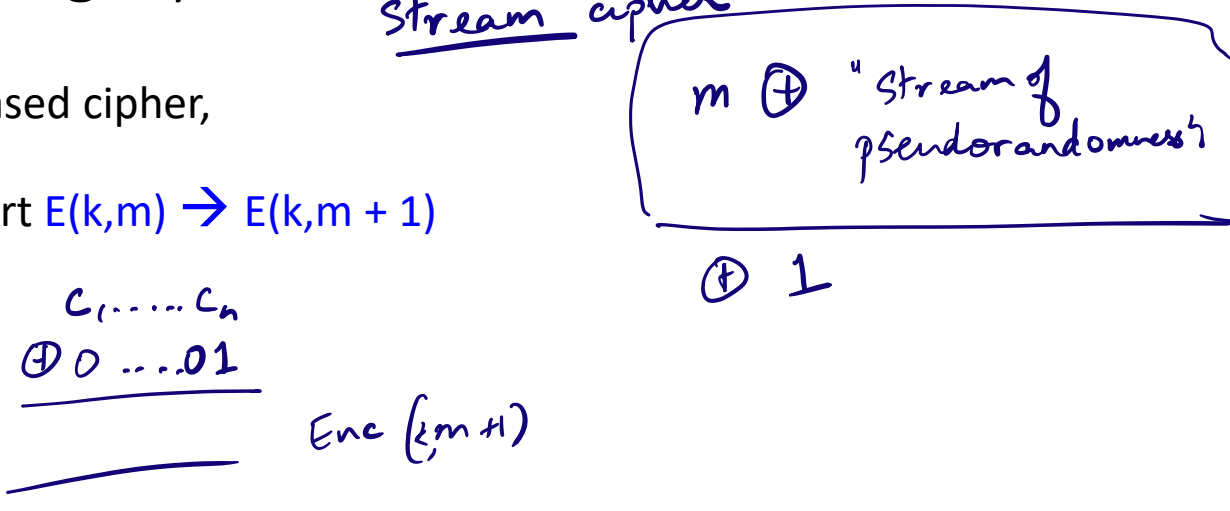
$\Rightarrow$ Contradiction

$\Rightarrow \mathcal{D}_1 \approx_c \mathcal{D}_2$.

Attack: Integrity

In the PRG-based cipher,

easy to convert E(k,m) → E(k,m + 1)



Example: RC4 Cipher (deprecated)

- Expand 128-bit seed to 2048 bits of pseudorandomness
- Use pseudorandomness to initialize internal state
- Used in HTTPS, WEP
- Weaknesses:

1. Not pseudorandom: e.g. [0,0] appears more often than it should
2. Related key attacks make it possible to recover the key

https://blog.cryptographyengineering.com/2013/03/12/attack-of-week-rc4-is-kind-of-broken-in/