Псевдослучайные функции

Лекция N 10 курса
“Современные задачи криптографии”
СПбГУ — SPRINT Lab

Юрий Лифшиц
yura@logic.pdmi.ras.ru

Лаборатория мат. логики ПОМИ РАН

Осень’2005

Каждый, кто рассматривает арифметические методы построения случайный чисел, без сомнения, совершает грех.

Джон фон Нейман

План лекции

1. Предсказание следующего бита
2. Псевдослучайные функции
3. Стойкость против восстановления ключа
4. Задачи

План лекции

1. **Предсказание следующего бита**
2. Псевдослучайные функции
3. Стойкость против восстановления ключа
4. Задачи

Вспомним определение

Функция $G : X \to Y$ называется псевдослучайным генератором, если

Вспомним определение

Функция $G : X \to Y$ называется псевдослучайным генератором, если $G(\mathcal{U}_X)$ и $\mathcal{U}_Y$ — вычислительно неразличимы ($\mathcal{U}_X, \mathcal{U}_Y,$ — равномерные распределения)

Вспомним определение

Функция $G : X \to Y$ называется псевдослучайным генератором, если $G(\mathcal{U}_X)$ и $\mathcal{U}_Y$ — вычислительно неразличимы ($\mathcal{U}_X, \mathcal{U}_Y,$ — равномерные распределения)

Неформально, генератор должен проходить все полиномиальные тесты на случайность

Вспомним определение

Функция $G : X \to Y$ называется псевдослучайным генератором, если $G(\mathcal{U}_X)$ и $\mathcal{U}_Y$ — вычислительно неразличимы ($\mathcal{U}_X, \mathcal{U}_Y,$ — равномерные распределения)

Неформально, генератор должен проходить все полиномиальные тесты на случайность

Вопрос: есть ли какой-нибудь конкретный тест, из которого следует случайность по всем остальным?

Предсказание следующего бита

Определение: распределение D проходит тест на прдсказание следующего бита, если для любого полиномиального алгортма A и любого p верно

$$\Pr_{t \in D}[A(t_1, \ldots, t_p) = t_{p+1}] < \frac{1}{2} + \nu(|t|)$$

Предсказание следующего бита

Определение: распределение D проходит тест на прдсказание следующего бита, если для любого полиномиального алгортма A и любого p верно

$$\Pr_{t \in D}[A(t_1, \ldots, t_p) = t_{p+1}] < \frac{1}{2} + \nu(|t|)$$

Говорят, что генератор G проходит тест на следующий бит, если распределение его выходов проходит этот тест.

Теорема о следующем бите

Теорема

Функция $G : \{0,1\}^n \to \{0,1\}^N$ *проходит тест на следующий бит IFF она является псевдослучайным генератором (проходит все полиномиальные тесты).*

Теорема о следующем бите

Теорема

Функция $G : \{0,1\}^n \to \{0,1\}^N$ *проходит тест на следующий бит IFF она является псевдослучайным генератором (проходит все полиномиальные тесты).*

Доказательство.

$\Leftarrow$: очевидно

Теорема о следующем бите

Теорема

Функция $G : \{0,1\}^n \to \{0,1\}^N$ *проходит тест на следующий бит IFF она является псевдослучайным генератором (проходит все полиномиальные тесты).*

Доказательство.

$\Leftarrow$: очевидно

$\Rightarrow$: докажем, что если G проваливает какой-то полиномиальный тест A', то G проваливает и некоторый тест A' на предсказание следующего бита.

Теорема о следующем бите

Теорема

Функция $G : \{0,1\}^n \to \{0,1\}^N$ *проходит тест на следующий бит IFF она является псевдослучайным генератором (проходит все полиномиальные тесты).*

Доказательство.

$\Leftarrow$: очевидно

$\Rightarrow$: докажем, что если G проваливает какой-то полиномиальный тест A', то G проваливает и некоторый тест A' на предсказание следующего бита.

Будем пользоваться гибридным методом! $\square$

Продолжение доказательства

У нас есть распределение G_N, порожденное G.

Продолжение доказательства

У нас есть распределение G_N, порожденное G.

Определим G'_k как распределение префиксов длины k на выходах G. Будем случайно и равновероятно дописывать каждый префикс случайными битами до длины N. Получим распределение G_k.

Продолжение доказательства

У нас есть распределение G_N, порожденное G.

Определим G'_k как распределение префиксов длины k на выходах G. Будем случайно и равновероятно дописывать каждый префикс случайными битами до длины N. Получим распределение G_k.

Наблюдение: $G_0 = U$, и при росте k распределение постепенно становится все более “псевдослучайным”.

Продолжение доказательства

У нас есть распределение G_N, порожденное G.

Определим G'_k как распределение префиксов длины k на выходах G. Будем случайно и равновероятно дописывать каждый префикс случайными битами до длины N. Получим распределение G_k.

Наблюдение: $G_0 = U$, и при росте k распределение постепенно становится все более “псевдослучайным”.

Применяем гибридный метод: алгоритм A различает G_0 и G_n, значит для какого-то k он различает G_k и G_{k+1}!

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$
2. Выписывает строки $T_0 = t_1, \ldots, t_k, 0, r_{k+2}, \ldots, r_N$ и $T_1 = t_1, \ldots, t_k, 1, r_{k+2}, \ldots, r_N$

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$
2. Выписывает строки $T_0 = t_1, \ldots, t_k, 0, r_{k+2}, \ldots, r_N$ и $T_1 = t_1, \ldots, t_k, 1, r_{k+2}, \ldots, r_N$
3. Вычисляет $a_0 = A(T_0)$ и $a_1 = A(T_1)$

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$
2. Выписывает строки $T_0 = t_1, \ldots, t_k, 0, r_{k+2}, \ldots, r_N$ и $T_1 = t_1, \ldots, t_k, 1, r_{k+2}, \ldots, r_N$
3. Вычисляет $a_0 = A(T_0)$ и $a_1 = A(T_1)$
4. Если $a_0 = 0, a_1 = 1$, выдает "$t_{k+1} = 1$"

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$
2. Выписывает строки $T_0 = t_1, \ldots, t_k, 0, r_{k+2}, \ldots, r_N$ и $T_1 = t_1, \ldots, t_k, 1, r_{k+2}, \ldots, r_N$
3. Вычисляет $a_0 = A(T_0)$ и $a_1 = A(T_1)$
4. Если $a_0 = 0, a_1 = 1$, выдает "$t_{k+1} = 1$"
5. Если $a_0 = 1, a_1 = 0$, выдает "$t_{k+1} = 0$"

Окончание доказательства

Пусть $A = 1$ чаще на G_{k+1}

Тест A':

1. Получает биты $t_1, \ldots, t_k$
2. Выписывает строки $T_0 = t_1, \ldots, t_k, 0, r_{k+2}, \ldots, r_N$ и $T_1 = t_1, \ldots, t_k, 1, r_{k+2}, \ldots, r_N$
3. Вычисляет $a_0 = A(T_0)$ и $a_1 = A(T_1)$
4. Если $a_0 = 0, a_1 = 1$, выдает "$t_{k+1} = 1$"
5. Если $a_0 = 1, a_1 = 0$, выдает "$t_{k+1} = 0$"
6. Если $a_0 = a_1$, выдает случайный ответ

План лекции

1. Предсказание следующего бита
2. **Псевдослучайные функции**
3. Стойкость против восстановления ключа
4. Задачи

Псевдослучайные функции

Функция $F : K \times D \longrightarrow R$ называется псевдослучайной, если для любого полиномиального противника A выполнено:

$$|Pr[A^{F_k(\cdot)} = 1] - Pr[A^{R(\cdot)} = 1]| < \nu(log|D|)$$

Конструкция

Теорема

Если псевдослучайные генераторы существуют, то псевдослучайные функции тоже.

Конструкция

Теорема

Если псевдослучайные генераторы существуют, то псевдослучайные функции тоже.

Доказательство.

Пусть $G(x) = G_1(x) \parallel G_2(x)$, генератор $B^n \to B^{2n}$.

Конструкция

Теорема

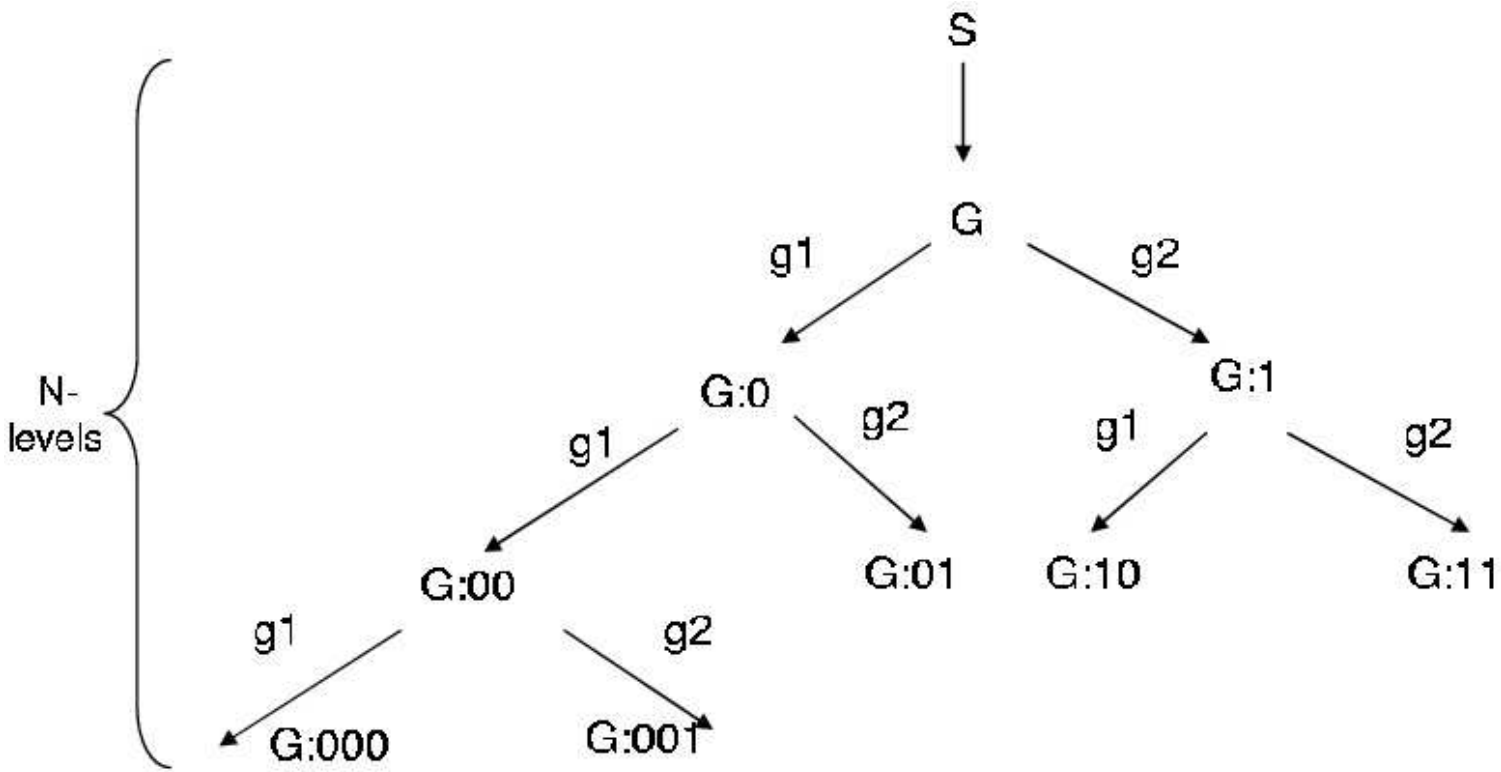
Если псевдослучайные генераторы существуют, то псевдослучайные функции тоже.

Доказательство.

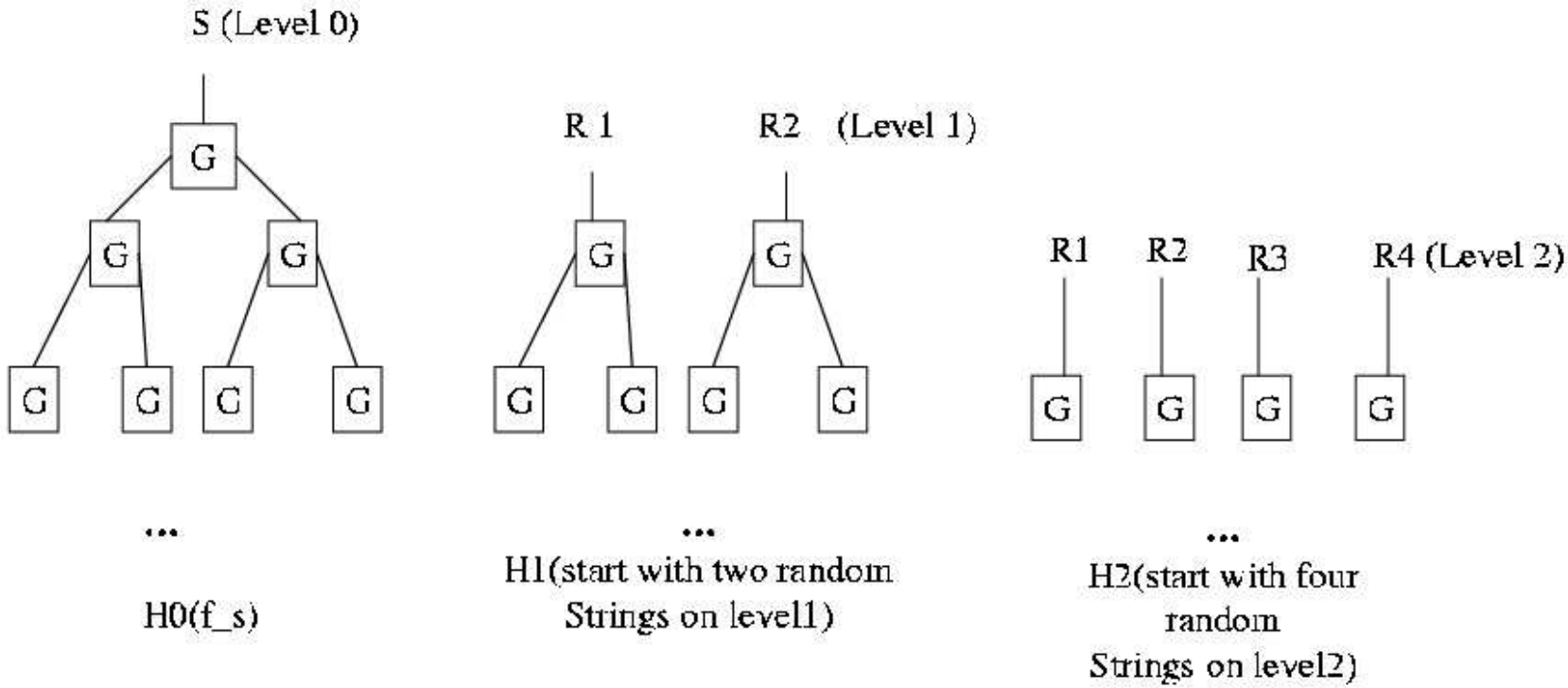
Пусть $G(x) = G_1(x) \parallel G_2(x)$, генератор $B^n \to B^{2n}$.

Тогда $F_s(x) = G_{x_1}(G_{x_2}(\dots G_{x_n}(s)\dots))$ — псевдослучайная функция. □

Конструкция



Гибридный метод



Рассмотрим промежуточные функции и докажем, что если можно различит две соседних, то можно взломать и исходный генератор G

План лекции

1. Предсказание следующего бита
2. Псевдослучайные функции
3. **Стойкость против восстановления ключа**
4. Задачи

Восстановление ключа

Семейство функций F_k называется стойким против восстановления ключа, если для любого полиномиального противника

$$Pr_{k \in K}[A^{F_k} = k] < \nu(|k|)$$

Теорема о стойкости

Теорема

Псевдослучайные функции обладают стойкостью против восстановления ключа.

Теорема о стойкости

Теорема

Псевдослучайные функции обладают стойкостью против восстановления ключа.

Доказательство.

TBD □

План лекции

1. Предсказание следующего бита
2. Псевдослучайные функции
3. Стойкость против восстановления ключа
4. **Задачи**

Задачи

Постройте из псевдослучайного генератора $G : B^n \to B^N$ псевдослучайную функцию $F : n \times \log N \to \{0, 1\}$

Задачи

Постройте из псевдослучайного генератора $G : B^n \to B^N$ псевдослучайную функцию $F : n \times \log N \to \{0, 1\}$

Докажите, что если существуют псевдослучайные функции, то псевдослучайные генераторы тоже существуют

Последний слайд

Если не запомните ничего другого:

- Псевдослучайная функция — это **семейство** функций, случайного представителя которой нельзя отличить от случайной функции

Последний слайд

Если не запомните ничего другого:

- Псевдослучайная функция — это **семейство** функций, случайного представителя которой нельзя отличить от случайной функции
- Псевдослучайную функцию можно построить из генератора

Последний слайд

Если не запомните ничего другого:

- Псевдослучайная функция — это **семейство** функций, случайного представителя которой нельзя отличить от случайной функции
- Псевдослучайную функцию можно построить из генератора
- Псевдослучайные функции обладают стойкостью против извлечения ключа

Последний слайд

Если не запомните ничего другого:

- Псевдослучайная функция — это **семейство** функций, случайного представителя которой нельзя отличить от случайной функции
- Псевдослучайную функцию можно построить из генератора
- Псевдослучайные функции обладают стойкостью против извлечения ключа

Последний слайд

Если не запомните ничего другого:

- Псевдослучайная функция — это **семейство** функций, случайного представителя которой нельзя отличить от случайной функции
- Псевдослучайную функцию можно построить из генератора
- Псевдослучайные функции обладают стойкостью против извлечения ключа

Вопросы?