Прикладная Криптография: Симметричные криптосистемы Поточные шифры

Макаров Артём

МИФИ 2024

Тест.

Пусть задана игра на семантическую стойкость для алгоритма A против шифра $\mathrm{E} = (E, D)$ на (K, M, C).

W_b - событие того, что $b' = 1$ в эксперименте b
W - событие, при котором $b' = b$

- Положить телефон экраном вниз справа от себя
- Не разговаривать с соседями
- Не пользоваться конспектами и электронными устройствами
- Написать номер (по таблице) и ФИО на листочке
- Написать краткий ответ на вопрос
- Дождаться окончания теста

Тест.

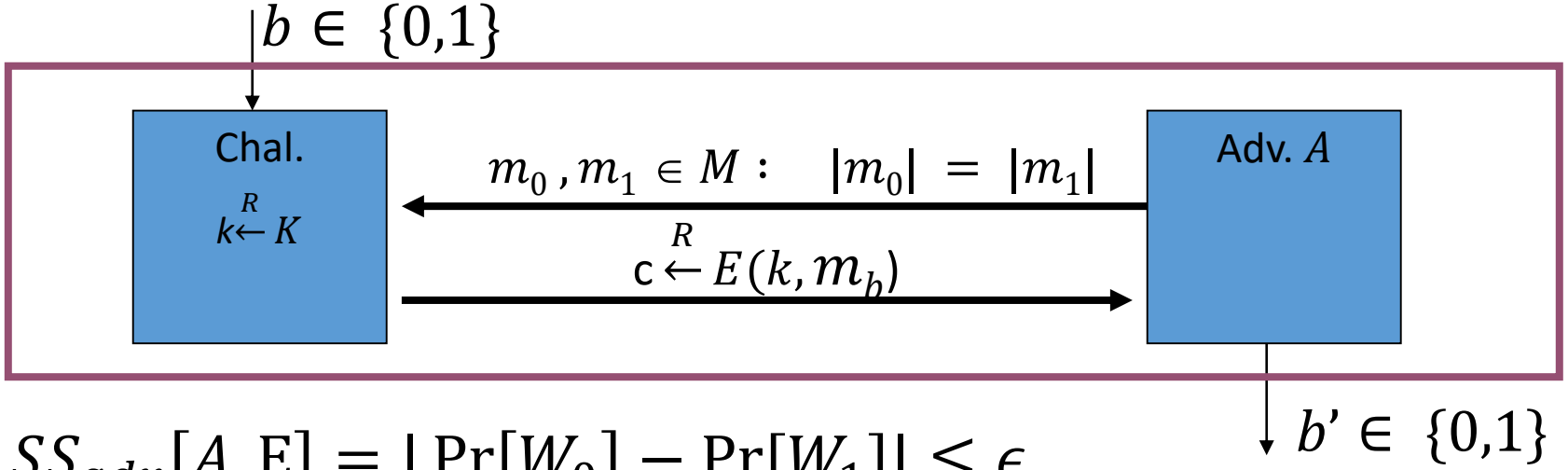
Пусть задана игра на семантическую стойкость для алгоритма A против шифра $\mathrm{E} = (E, D)$ на (K, M, C).

W_b - событие того, что $b' = 1$ в эксперименте b
W - событие, при котором $b' = b$

- Что отправляет и что получает противник?
- $\mathrm{SSadv}^*[A, \mathrm{E}] =$ **???** (функция от W)

TIME IS UP

Семантическая стойкость



$$SS_{adv}[A, \mathrm{E}] = |\Pr[W_0] - \Pr[W_1]| \leq \epsilon$$

- ϵ – **пренебрежимо малая величина**.
- Претендент и Противник – **эффективные алгоритмы**

Пренебрежимо малые величины

Функция $f: Z_{\geq 1} \to R$ **называется пренебрежимо малой (negligible)**, если для всех c $\in R_{>0}$ $\exists n_0 \in Z_{\geq 1}: \forall n \geq n_0$ справедливо неравенство:

$$|f(n)| < \frac{1}{n^{\mathrm{c}}}.$$

Теорема 2.1. Функция $Z_{\geq 1} \to R$ пренебрежимо малая, тогда и только тогда когда $\forall c > 0$ справедливо равенство:

$$\lim_{n \to \infty} f(n) n^c = 0.$$

Т.е. на бесконечности функция от n убывает быстрее любого полинома от n .

Примеры

- **Пренебрежимо малые функции**: $2^{-n}, 2^{-\sqrt{n}}, n^{-\log n}$.
  - Убывают быстрее любых полиномов

- **Не пренебрежимо малые** функции: $n^2, n^{-2}, n^{-10000000}$.
  - $f(n) = n^2 : \exists c = 0, \forall n > 0 : \lim_{n\to\infty} f(n)n^c = n^2 * 1 \neq 0.$
  - $f(n) = n^{-2} : \exists c = 3, \forall n > 0 : \lim_{n\to\infty} f(n)n^c = n^{-2} * n^3 = n \neq 0.$
  - $f(n) = n^{-10000000} : \exists c = 10000001, \forall n > 0 : \lim_{n\to\infty} f(n)n^c = n \neq 0.$

Сверх-полиномиальные и полиномиально ограниченные функции

- Функция $f: Z_{\geq 1} \to R$ называется **сверх-полиномиальной (super-poly)**, если $1/f$ – пренебрежимо малая.
  - Растёт быстрее любого полинома на бесконечности.

- Функция $f: Z_{\geq 1} \to R$ называется **полиномиально-ограниченной (poly bounded)**, если $\exists c, d \in R_{\geq 0}: \forall n \geq 1$ имеет место неравенство:

$$|f(n)| \leq n^c + d$$

  - Может быть ограничена на бесконечности сверху полиномом степени c.

Пренебрежимо малые величина на практике

На практике величина ϵ – скаляр (формально – функция от некоторых фиксированных ранее параметров системы). Её «малость» оценивают исходя из необходимой для системы стойкости.

Пример:

- ϵ – не пренебрежимо малая, если событие вероятно произойдёт при обработки данных порядка гигабайта, $\epsilon \geq 1/2^{30}$
- ϵ- пренебрежимо малая, если событие вряд ли произойдёт при «жизни» ключа длины 160 бит, $\epsilon \leq 1/2^{80}$

Пренебрежимо малые величина на практике

При доказательстве стойкости часто получается формула преимущества, ограниченная сверху функцией от некоторых параметров. Пример:

$Adv_{SS}[A, \mathrm{E}] \leq q/2^k$, где q – максимально число зашифрований, k- длина ключа.

Для использования конкретной реализации нужно выбрать параметры q и k при заданном уровне стойкости.

Пусть хотим $Adv_{SS}[A, \mathrm{E}] \leq 1/2^{80}$, тогда для зашифрования, при использовании ключа с длиной $k = 128$ бит мы можем зашифровать $q \leq \frac{1}{2^{80-k}} = 2^{48}$ сообщений, при параметре стойкости 80 бит.

Параметры системы

Ранее, при введении понятия (вычислимого) шифра, мы описывали его без явного описания параметров.

На практике многие шифры и другие примитивы имеют так называемые **параметры системы**, влияющие на производительность и стойкость системы.

Пример – длина ключа (и максимального сообщения) в одноразовом блокноте, модуль и длина ключа в аддитивном одноразовом блокноте.

Эффективный алгоритм

Пусть λ – некоторый параметр. Пусть $p(\lambda)$ - полином над $Z_{\geq 1}$. Пусть A: $A(\lambda, x)$: $\lambda \in Z_{\geq 1}, x \in \{0,1\}^{\leq p(\lambda)}$ (т.е. длина вектора x полиномиальной ограничена на основе параметра).

Алгоритм A называется **эффективным**, если $\exists t(\lambda)$: $t -$ полиномиально ограниченная, $\epsilon(\lambda)$: $\epsilon -$ перенебрежимо малая: $\forall \lambda \in Z_{\geq 1}, \forall x \in \{0,1\}^{\leq p(\lambda)}$ вероятность того, что время исполнения алгоритма A на входе (λ, x) превысит $t(\lambda)$ не превосходит $\epsilon(\lambda)$.

Иными словами, алгоритм A – **эффективный**, если при заданном **параметре на полиномиально-ограниченном входе он почти всегда (т.е. за исключением конечного малого числа точек) выполняется за полиномиальное время**.

Пример эффективного алгоритма с параметром

Одноразовый блокнот переменной длины. $\mathrm{E} = (E, D)$ на $K = \{0,1\}^L, M = C = \{0,1\}^{\leq L}$, где $L = \lambda$ – фиксированный **параметр**

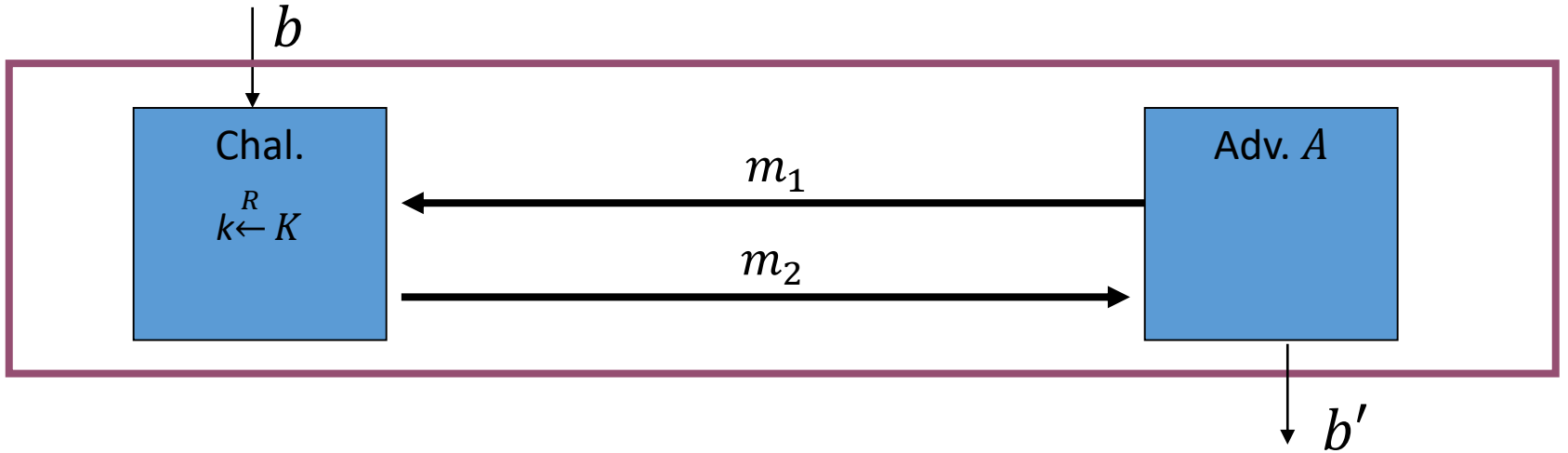
$$E(k, m) = k[0..l-1] \oplus m$$
$$D(k, c) = k[0..l-1] \oplus c$$

Длина входов алгоритма E: $in \in I = K \times M$: $|I| \leq 2^{2L} \Rightarrow |in| = 2L$, **полиномиально-ограниченна** сверху полиномом $p(l) = p(\lambda) = 2L + 1$

Время выполнения E: $\mathrm{t} \leq L$, **полиномиально-ограниченно** сверху полиномом $p(l) = p(\lambda) = L + 1$

Эффективность в игре

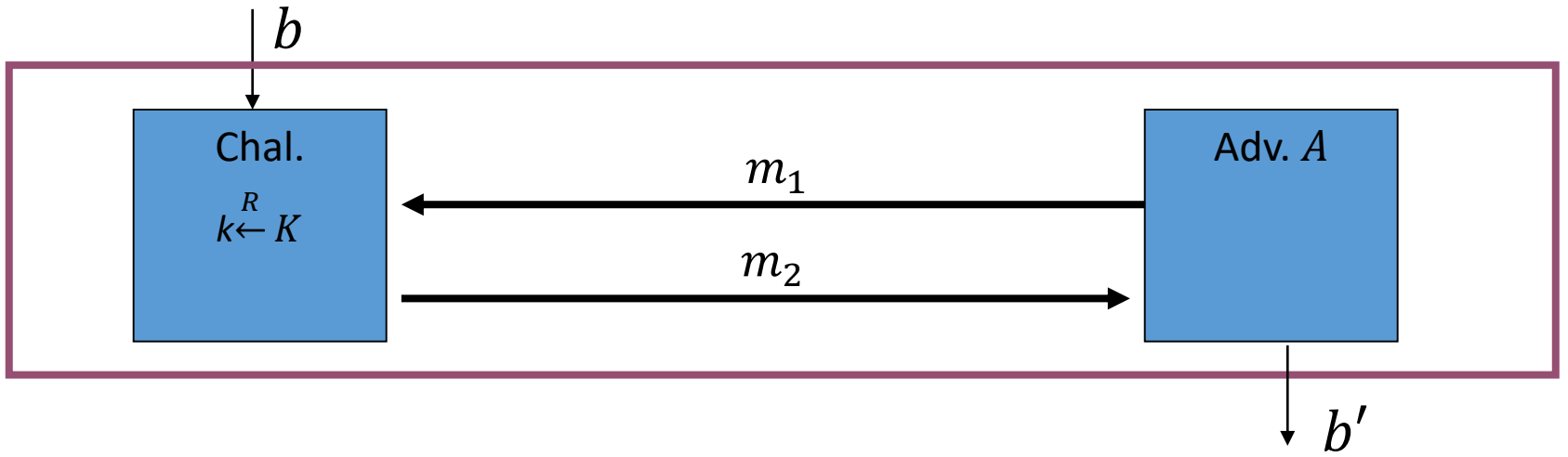
Ранее мы указывали, что в играх будем рассматривать только эффективные (вычислимые) алгоритмы, как для Претендента, так и для Противника. Иными словами, в игре должно быть полиномиально-ограниченное число шагов, Противник обладает полиномиально-ограниченным временем и памятью. Т.е. алгоритм игры должен быть эффективным.



Эффективность в игре

Два **эксперимента** игры называются **статистически неразличимыми**, если не существует эффективного алгоритма противника, способного различить эти эксперименты.

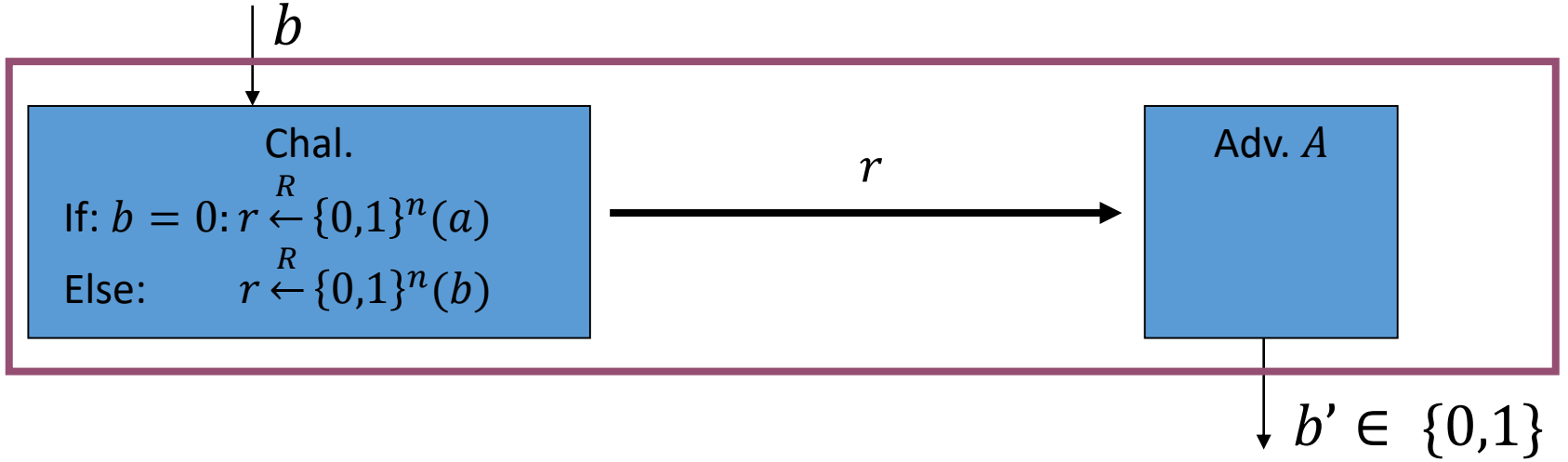
Т.е. $\forall A\ Adv_{disc} = |\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| \leq \epsilon$, где ϵ – пренебрежимо малая величина.



Эффективность в игре

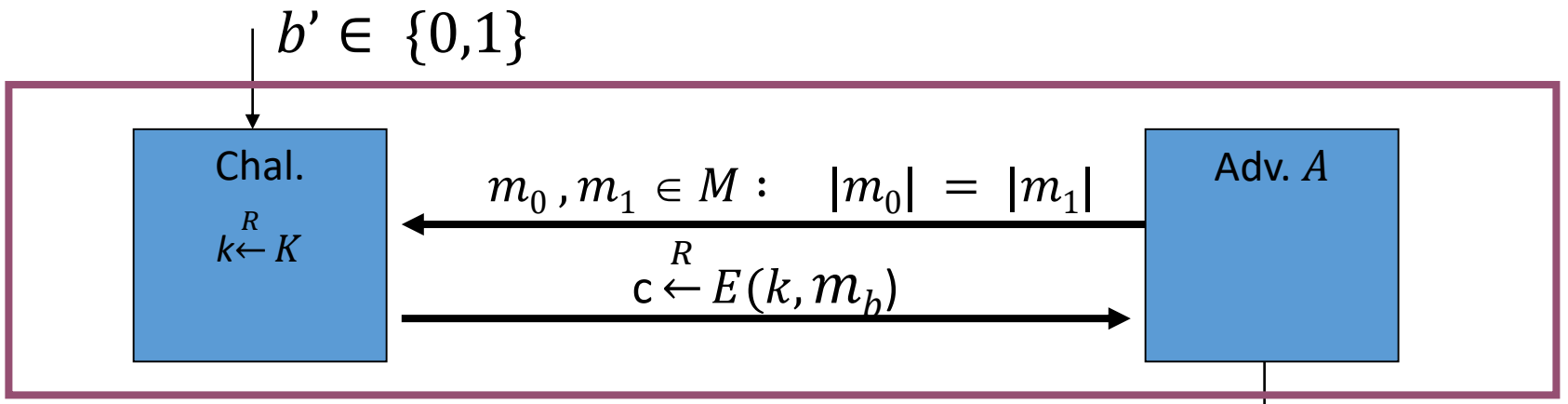
Пусть a, b – **распределения** на $\{0,1\}^n$. a и b **называются статистически неразличимыми**, если не существует эффективного алгоритма противника, способного различить эти распределения в игре на распознание. Обозначается $a \approx_p b$.

Т.е. $\forall A \; Adv_{disc} = |\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| \leq \epsilon$, где ϵ – пренебрежимо малая величина.



Альтернативная трактовка понятия абсолютной и семантической стойкости

Противник не может различить шифртексты двух выбранных сообщений



$$SS_{adv}[A, \mathrm{E}] = |\Pr[W_0] - \Pr[W_1]| \leq \epsilon$$

Параметр стойкости

Параметром стойкости называют двоичный логарифм, от необходимого числа операций для осуществления теоретической или практической атаки.

Пример: идеальный (нет атак, помимо перебора ключа) шифр с ключом длины l, параметр стойкости l бит (необходимо перебрать весь ключ).

Пример: Семантически стойкий шифр $\mathrm{E} = (E, D)$: $\forall A\ Adv_{SS}[A, E] \leq 1/2^k$, параметр стойкости k бит.

Оценки величин

Параметр стойкости 10 бит это много или мало? А 2^{10} бит? При каком параметре стойкости принято считать систему стойкой?

Необходимый параметр стойкости зависит от приложения используемой криптосистемы.

Для систем общего назначения рекомендуемые параметры стойкости 80-256 бит.

Оценки величин

$\sim 2^{240}$ - число элементарных частиц в обозримой вселенной

$\sim 1/2^{119}$- шанс выиграть в лотерею, с миллионом участников 6 раз подряд

$\sim 2^{60}$ - секунд с большого взрыва (2^{200} - в планковских единицах)

$\sim 2^{42}$ - вычислительная сложность майнинга биткоина (2018 год)

$\sim 2^{30}$ - можно перебрать на домашнем компьютере за несколько часов

Идея одноразового блокнота

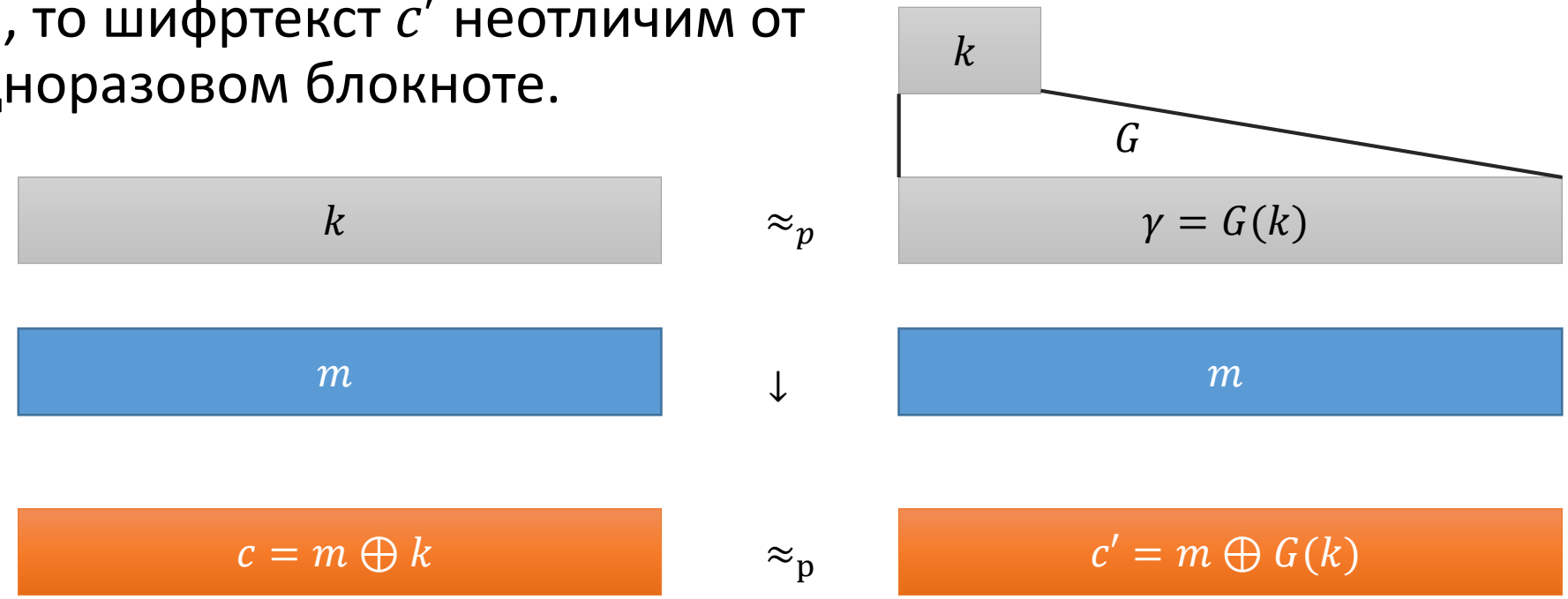
Одноразовый блокнот – сложение (побитное) случайного равновероятного вектора ключа с вектором открытого текста, для получения шифртекста.

Проблема (**Теорема Шеннона**) – длина (энтропия) ключа должна быть больше или равна длине сообщения.

Основная идея – заменить случайный длинный вектор ключа на «псевдослучайную» последовательность, называемую гаммой.

Идея одноразового блокнота

Заменяем использование случайного ключа k псевдослучайной последовательностью γ. Если последовательность «неотличима» от случайной равновероятной, то шифртекст c' неотличим от шифртекста в одноразовом блокноте.



Поточный шифр

Эффективно вычислимая функция $G: S \to R$ называется **псевдослучайным генератором** на (S, R) **PRG**.

Шифр $\mathrm{E} = (E, D)$ с параметрам (l, L) на (K, M, C): $K = \{0,1\}^l, M = C = \{0,1\}^L$, называется **поточным шифром**, если

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

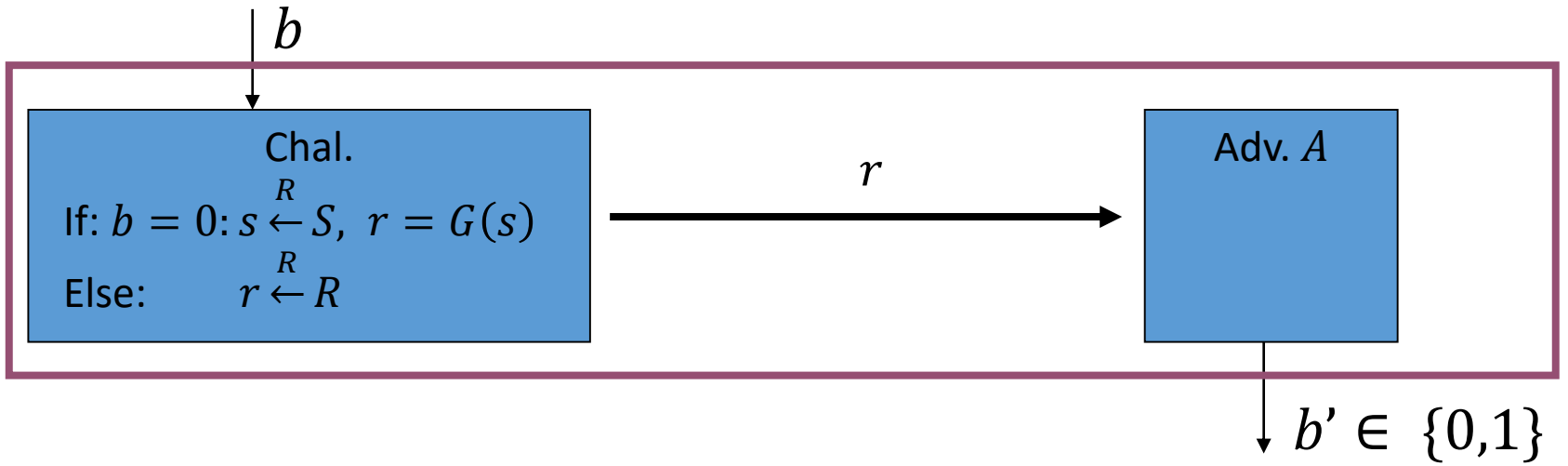
где $G: \{0,1\}^l \to \{0,1\}^L$ - псевдослучайный генератор.

Аналогично можно ввести Поточный шифр по произвольному модулю.

Стойкость поточного шифра сводится к «качеству» псевдослучайной последовательности $\gamma = G(k)$

Стойкий псевдослучайный генератор

Пусть G псевдослучайный генератор на (S, R). Рассмотрим игру с двумя экспериментами. В эксперименте 0 Претендент отправляет псевдослучайную величину $r = G(s)$. В эксперименте 1 – случайную величину $r \xleftarrow{R} R$. Задача Противника угадать, случайную, или псевдослучайную величину он получил.

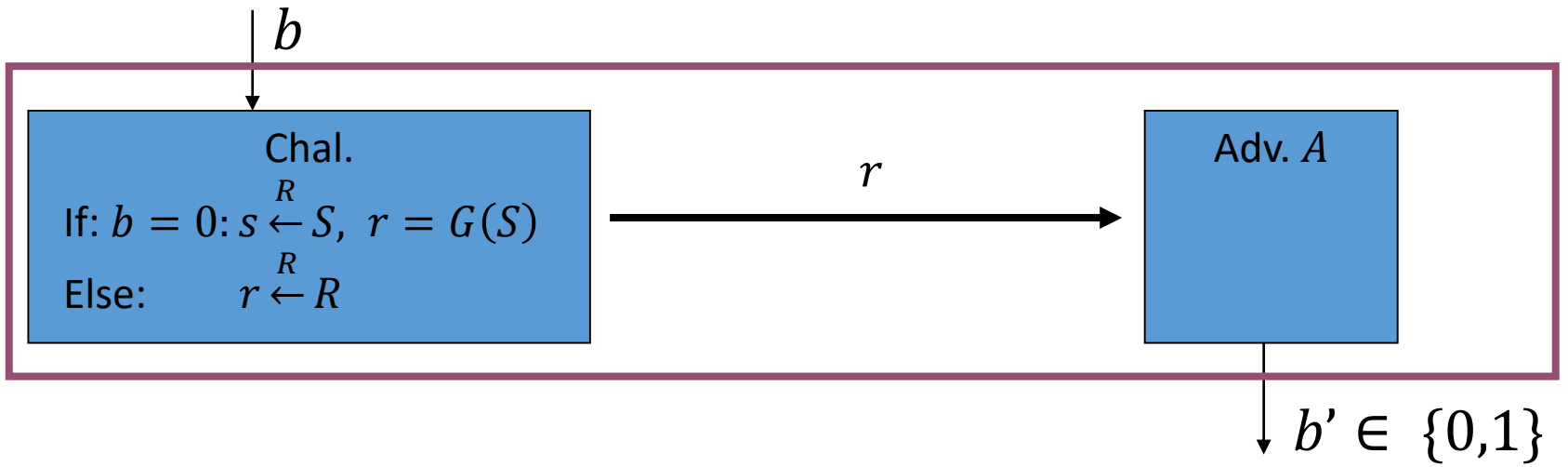


Стойкий псевдослучайный генератор

Пусть W_b - событие того, что $b' = 1$ в эксперименте b.

Тогда Преимуществом Противника A против алгоритма G в игре на различимость есть величина

$$Adv_{PRG}[A, G] = |\Pr[W_0] - \Pr[W_1]|.$$

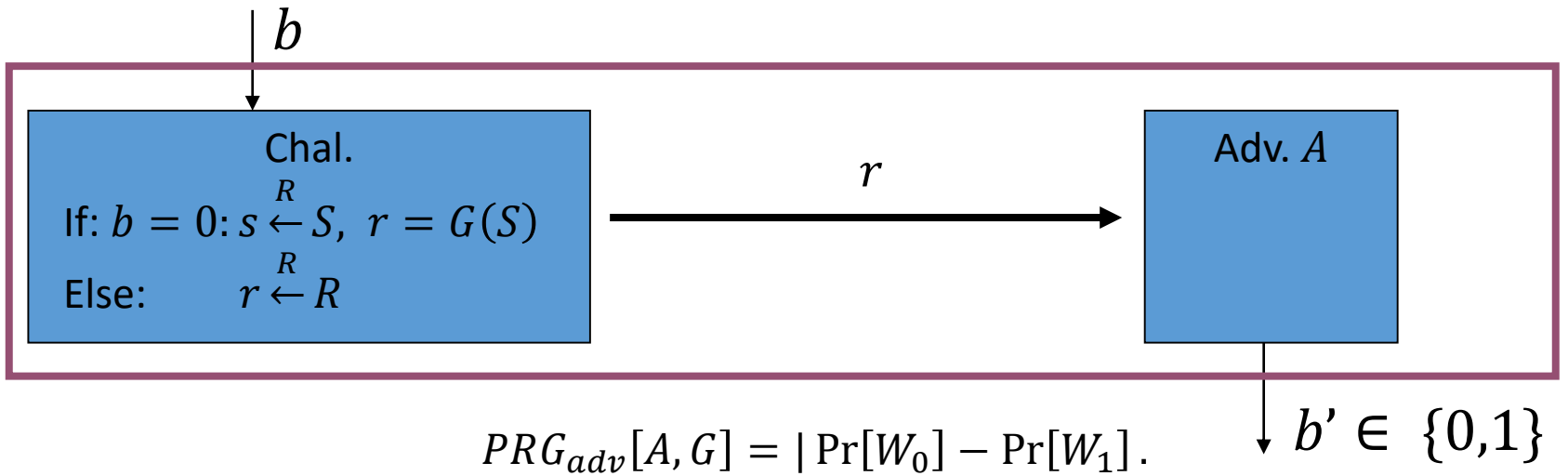


Стойкий псевдослучайный генератор

Генератор G называют **стойким псевдослучайным генератором** (secure PRG), если для любых эффективных противников A величина $PRG_{adv}[A, G] \leq \epsilon$, где ϵ – пренебрежимо малая величина.

Противника A часто называют **статистическим тестом**.

Если генератор G – стойкий, то последовательность $\gamma = G(s)$ называют (эффективно) **статистически неразличимой от случайной последовательности** или **стойкой псевдослучайной последовательностью**. Обозначается $\gamma \approx_P r$, где r – случайная последовательность.



$$PRG_{adv}[A, G] = |\Pr[W_0] - \Pr[W_1]|.$$

Энтропия генератора

Пусть G на $(S, \{0,1\}^n)$ - генератор.

Очевидно, что генератор может выдать не более $|S|$ различных последовательностей.

Часто в генераторах величина $\boldsymbol{s} \in_R S$ является начальным заполнением внутреннего состояния. Тогда максимально возможная энтропия выходной последовательности $\boldsymbol{\gamma} \leftarrow G(\boldsymbol{s}), \boldsymbol{s} \overset{R}{\leftarrow} S$ равна энтропии случайной величины $\boldsymbol{s}$, т.е.

$$H(\boldsymbol{\gamma}) \leq H(s) \leq \log_2 |S|$$

Таким образом максимально возможная длина периода генератора

$$l \leq 2^{H(\boldsymbol{\gamma})} \leq |S|$$

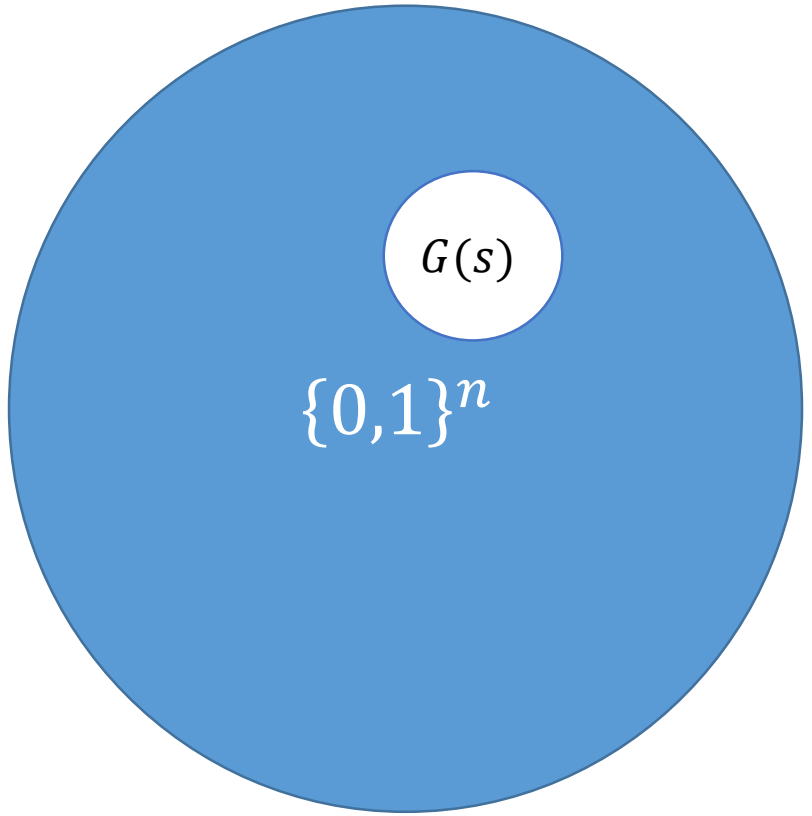
Пример: Пусть $S = \{0,1\}^{128}$ - множество состояний и ключей, тогда максимально возможный период выходной последовательности не превышает 2^{128}, энтропия 128 бит.

Статистическая неразличимость

Пусть G на $(S, \{0,1\}^n)$.

Рассмотрим множество возможных значений $R \subset \{0,1\}^n$: $\{r = G(S), \forall s \in S\}$.

Тогда если G – стойкий генератор, то эффективный Противник не может определить содержится ли элемент $r' \in \{0,1\}$ в R.



Непредсказуемость генераторов

Пусть G псевдослучайный генератор на (S, R).

- Генератор G называется **предсказуемым**, если ∃ эффективный алгоритм A и $\exists 0 \leq i \leq n-1$:

$$Adv_{pred} = |\Pr\left[A(G(k)[0..i])\right) = G(k)[i+1]] - 1/2| > \epsilon$$

Где ϵ – не пренебрежимо малая. Т.е. Существует эффективный алгоритм способный по $i+1$ биту предсказать $i+2$.

- Генератор G называется **непредсказуемым**, если ∀ эффективных алгоритмов справедливо

$$Adv_{pred} = |\Pr\left[A(G(k)[0..i])\right) = G(k)[i+1]] - 1/2| \leq \epsilon,$$

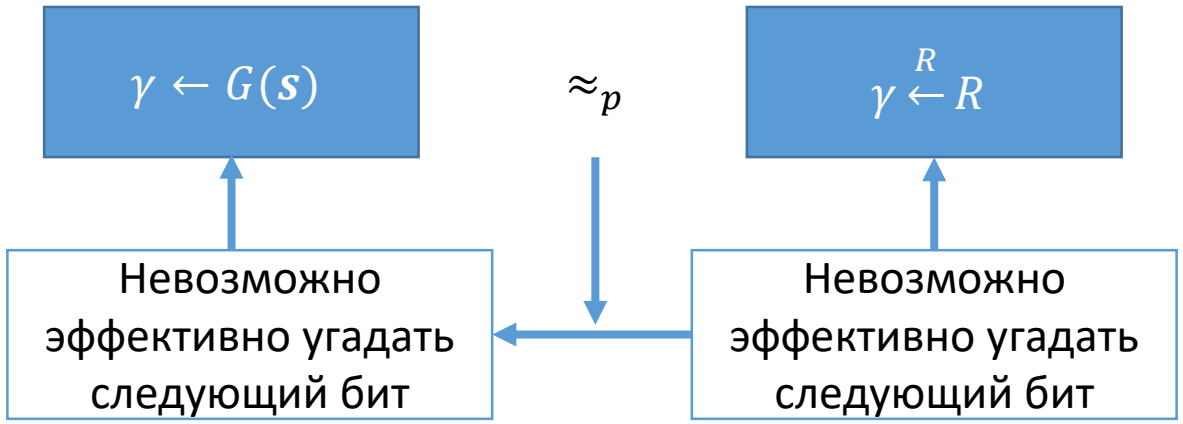
для пренебрежимо малой ϵ.

Непредсказуемость генераторов

Теорема 2.2. Пусть G псевдослучайный генератор (PRG) на (S, R).

Если G – стойкий, то G – непредсказуемый.

▷ без доказательства. Идея доказательства – если G – стойкий, то его выход вычислительно неотличим от случайной последовательности. А для случайной последовательности невозможно предсказать следующий бит. $Pred_{avd}[A, G] = PRG_{adv}[B, G]$ ◁



Непредсказуемость генераторов

Теорема 2.3. Yao’82 . Пусть G псевдослучайный генератор (PRG) на (S, R).

Если G – непредсказуемый, то G – стойкий

⊳ без доказательства. Идея доказательства – если мы не можем предсказать 1 следующий бит, то значит у нас нет никаких возможностей определить является ли данная величина случайной, или выходом псевдослучайного генератора ⊲

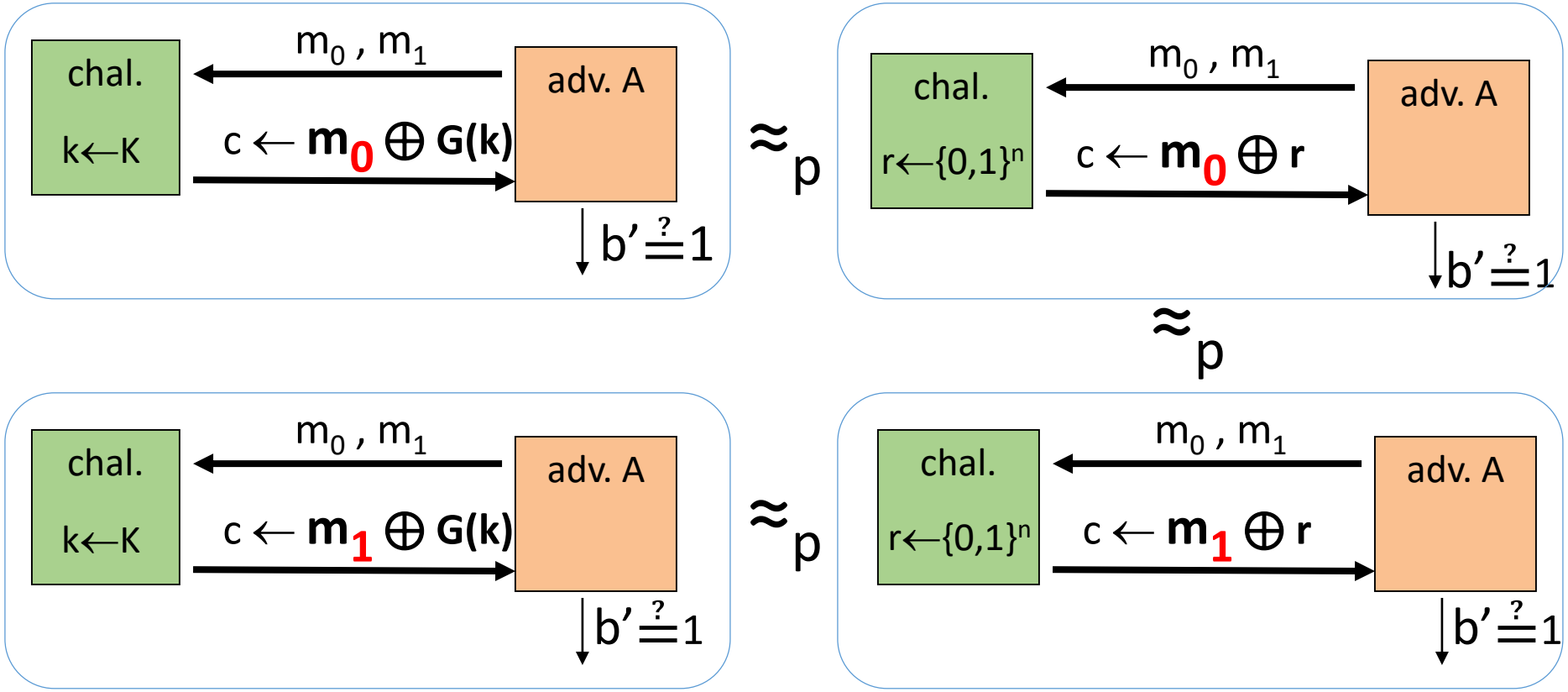
Поточные шифры и семантическая стойкость

Теорема 2.4. Пусть $G: S \to \{0,1\}^n$ стойкий генератор (PRG).

Тогда поточный шифр E определённый с использованием G семантически стойкий, т.е. $\forall A$: A – противник в игре на семантическую стойкость, $\exists$ противник B в игре на стойкость PRG (различимость):

$$Adv_{ss}[A, E] \leq 2 * Adv_{PRG}[B, G]$$

Идея доказательства



Теорема 2.4. Пусть $G: S \to \{0,1\}^n$ стойкий генератор (PRG).

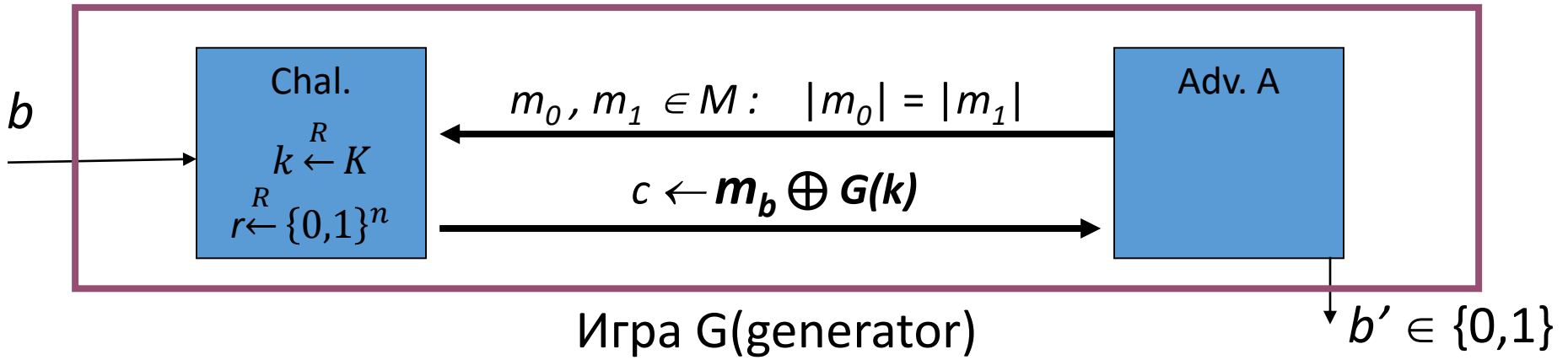
Тогда поточный шифр E определённый с использованием G семантически стойкий, т.е. $\forall A$: A – противник в игре на семантическую стойкость, $\exists$ противник B в игре на стойкость PRG (различимость):

$$Adv_{ss}[A, E] \leq 2 * Adv_{PRG}[B, G]$$

▷ Пусть A противник в игре на семантическую стойкость.

Пусть претендент также генерирует $r \overset{R}{\leftarrow} \{0,1\}^n$.

Пусть W_b событие, при котором $b' = 1$



Игра G(generator)

Теорема 2.4. Пусть $G: S \to \{0,1\}^n$ стойкий генератор (PRG).

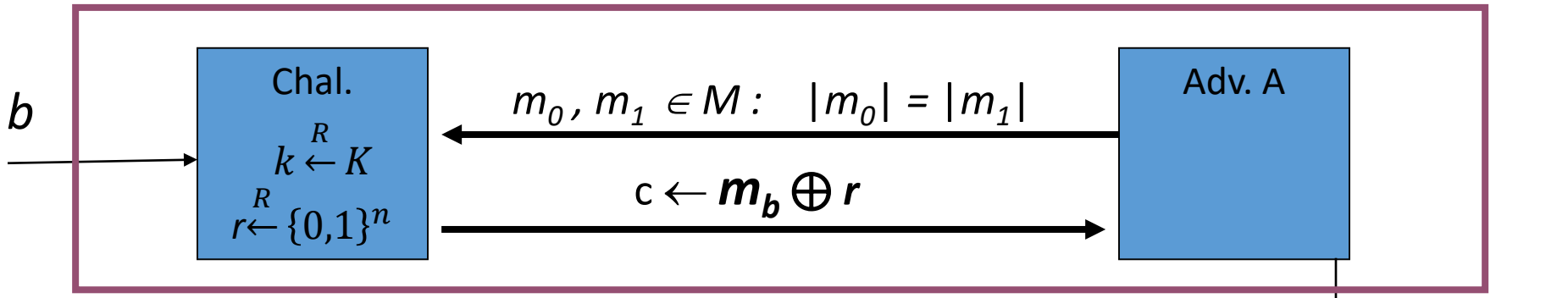
Тогда поточный шифр E определённый с использованием G семантически стойкий, т.е. $\forall A$: A – противник в игре на семантическую стойкость, $\exists$ противник B в игре на стойкость PRG (различимость):

$$Adv_{ss}[A, E] \leq 2 * Adv_{PRG}[B, G]$$

Пусть A противник в игре на семантическую стойкость.

Пусть претендент шифрует сообщение одноразовым блокнотом (OTP).

Пусть R_b событие, при котором $b' = 1$.



Игра R(random)

Теорема 2.4. Пусть $G: S \to \{0,1\}^n$ стойкий генератор (PRG).

Тогда поточный шифр E определённый с использованием G семантически стойкий, т.е. $\forall A$: A – противник в игре на семантическую стойкость, $\exists$ противник B в игре на стойкость PRG (различимость):

$$Adv_{ss}[A, E] \leq 2 * Adv_{PRG}[B, G]$$

Утверждение 2.4.1. $Adv_{ss}[A, OTP^*] = 0 = |\Pr[R_0] - \Pr[R_1]|$

Утверждение 2.4.2. $\exists B$: $Adv_{PRG}[B, G] = |\Pr[W_b] - \Pr[R_b]|$, т.е. B – противник, которые пытается различить PRG и OTP.

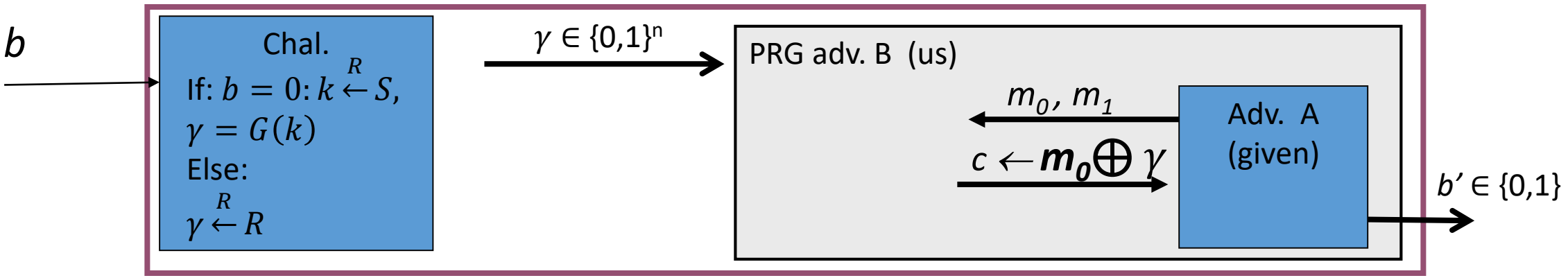
$$\Rightarrow \quad Adv_{SS}[A,E] = |\Pr[W_0] - \Pr[W_1]| \leq 2 * Adv_{PRG}[B,G]$$

* OTP – одноразовый блокнот

Поточные шифры и семантическая стойкость

Утверждение 2.4.2. $\exists B: Adv_{PRG}[B, G] = |\Pr[W_b] - \Pr[R_b]|$, т.е. B – противник, которые пытается различить PRG и OTP.

Алгоритм B:



$$Adv_{PRG}[B, G] = |\Pr[B(r) = 1] - \Pr[B(G(k)) = 1|, \text{ где } k \xleftarrow{R} K, r \xleftarrow{R} \{0,1\}^n$$

$$\Pr[B(r) = 1] = \Pr[R_0], \quad \Pr[B(G(k)) = 1| = \Pr[W_0]$$

$$\Rightarrow Adv_{PRG}[B, G] = |\Pr[W_b] - \Pr[R_b]|.$$

Поточные шифры и семантическая стойкость

Теорема 2.5. Пусть $G: S \to \{0,1\}^n$ генератор (PRG).

Тогда если поточный шифр E определённый с использованием G семантически стойкий, то G – стойкий генератор.

▻без доказательства ◅

Тест.

Пусть задана игра на семантическую стойкость для алгоритма A против шифра $\mathrm{E} = (E, D)$ на (K, M, C).

W_b - событие того, что $b' = 1$ в эксперименте b
W - событие, при котором $b' = b$

- $\mathrm{SSadv}[A, \mathrm{E}] =$ **???** (функция от $W_b, b = 0,1$)
- $\mathrm{SSadv}^*[A, \mathrm{E}] =$ **???** (функция от W)
- Связь $\mathrm{SSadv}[A, \mathrm{E}]$ и $\mathrm{SSadv}^*[A, \mathrm{E}]$