Обфускация программ: методы и приложения

В.А. Захаров,
Н.Н. Кузюрин, А.В. Шокуров

Институт системного программирования,
ф-т ВМиК МГУ им. М.В. Ломоносова

ОБФУСКАЦИЯ ПРОГРАММ

— это такая разновидность эквивалентных преобразований программ, которая предназначена для затруднения понимания программ, извлечения полезной информации об алгоритмах, структурах данных, секретных ключах, содержащихся в программах.

ОБФУСКАЦИЯ ПРОГРАММ

— это такая разновидность эквивалентных преобразований программ, которая предназначена для затруднения понимания программ, извлечения полезной информации об алгоритмах, структурах данных, секретных ключах, содержащихся в программах.

- **Первое упоминание:**
  Diffie W., Hellman M. New directions in cryptography, 1976.
- **Первая научная статья:**
  Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations,, 1997.
- **Первая математическая постановка задачи:**
  Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K. On the (Im)possibility of obfuscating programs, 2001.

НЕКОТОРЫЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ

1. Определения стойкости обфускации.
2. Обфускация с разделением программы.
3. Программы, непроницаемые для статического анализа.

СТОЙКОСТЬ ОБФУСКАЦИИ

Модель «черного ящика»

Вероятностный алгоритм $\mathcal{O}$ называется **обфускатором, стойким в модели «черного ящика»**, если он удовлетворяет следующим требованиям:

1. (функциональность) для любой машины Тьюринга M
$$\mathbf{M} \approx \mathcal{O}(\mathbf{M}).$$
2. (полиномиальное замедление) Существует такой полином $p(\cdot)$, что для любой машины Тьюринга M
$$\mathbf{size}(\mathcal{O}(\mathbf{M})) \leq \mathbf{p}(\mathbf{size}(\mathbf{M})),\ \mathbf{time}(\mathcal{O}(\mathbf{M})) \leq \mathbf{p}(\mathbf{time}(\mathbf{M})).$$
3. (стойкость) Для любой PPT A (противника) существует PPT S (симулятор) и пренебрежимо малая функция ν, такие что неравенство
$$|\mathbf{Pr}\{\mathbf{A}(\mathcal{O}(\mathbf{M}))=\mathbf{1}\} - \mathbf{Pr}\{\mathbf{S}^{\mathbf{M}}(\mathbf{1}^{\mathbf{size(M)}})=\mathbf{1}\}| \leq \nu(\mathbf{size}(\mathbf{M}))$$
выполняется для любой машины Тьюринга M.

СТОЙКОСТЬ ОБФУСКАЦИИ

Теорема

[Barak B., Goldreich O., el al., 2001]

Обфускаторов, стойких в модели «черного ящика», не существует .

СТОЙКОСТЬ ОБФУСКАЦИИ
Другие определения стойкости

- **стойкость в модели «серого ящика»**:
  Варновский Н.П., 2004.
- **обфускация с дополнительным входом**:
  Goldwasser S., Tauman Kalai Y.T., 2005.
- **наилучшая возможная обфускация**:
  Goldwasser S., Rothblum G.N., 2007.
- **обфускация алгоритмов**:
  Варновский Н.П., 2004.
- **защита констант (ключей)**:
  Варновский Н.П., 2004; Hofheinz D., Malobe-Lee J., Stam M. 2007.
- **обфускация предикатов**:
  Захаров В.А., Варновский Н.П., 2003; Варновский Н.П., 2004.

СТОЙКОСТЬ ОБФУСКАЦИИ

ПОЛОЖИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

Существует обфускатор,
стойко защищающий алгоритмы ,
представленные детерминированными конечными автоматами
(или OBDD полиномиального размера).

Кузюрин Н.П., Шокуров А.А. и др., 2007;
Goldwasser, S., Rothblum G., 2007.

ПОЛОЖИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

Точечной называется функция $f_{a,b} : \{0,1\}^n \to \{0,1\}^n$, $a, b \in \{0,1\}^n$, удовлетворяющая условию

$$f_{a,b}(x) = \begin{cases} b, \text{ если } x = a, \\ 0^n, \text{ если } x \neq a. \end{cases}$$

Рассмотрим семейство точечных функций $\mathcal{F}_n = \{f_{u,v} : u, v \in \{0,1\}^n\}$ и предикат $\pi_{a,b}(f) = (f \equiv f_{a,b})$.

Теорема [Canetti R. 1997; Захаров В.А., Варновский Н.П., 2003; Lynn B., Prabhakaran M., Sahai A. 2004, Wee H, 2005]

Если существуют односторонние перестановки, то предикат $\pi_{a,b}$, определенный на семействе программ, вычисляющих точечные функции $\mathcal{F}_n$, имеет стойкую обфускацию.

ПОЛОЖИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

Программа повторного шифрования принимает на вход шифр m_K сообщения m, зашифрованного при помощи ключа K, и вычисляет шифр m_R того же сообщения m, зашифрованного при помощи другого ключа R.

Теорема [Hohenberger S., Rothblum G. N., Shelat A., Vaikuntanathan V., 2007]

Существуют программы повторного шифрования, для которых можно построить обфускацию, стойкую в среднем в модели «черного ящика».

СТОЙКОСТЬ ОБФУСКАЦИИ

Пусть M — это программа с параметром (переменной) x. Обозначим M_c пример программы M, в которой вместо параметра x подставлена константа $c \in \{0,1\}^n$.

Вероятностный алгоритм $\mathcal{O}$ называется **обфускатором, скрывающим константу**, для параметризованного семейства программ $\mathcal{F} = \{M_c \ : \ c \in \{0,1\}^n, \ n \geq 1\}$, если он удовлетворяет следующим требованиям:

1. (функциональность)
2. (полиномиальное замедление)
3. (стойкость) Для любой PPT A (противника) существует PPT S (симулятор) и пренебрежимо малая функция ν, такие что неравенство

$$|\mathbf{Pr}\{\mathbf{A}[\mathcal{O}(\mathbf{M_{c_0}}), \mathbf{M_c}] = \mathbf{1}\} - \mathbf{Pr}\{\mathbf{S}^{\mathbf{M_{c_0}}}[\mathbf{1}^{\mathbf{size}(\mathbf{M_{c_0}})}, \mathbf{M_c}] = \mathbf{1}\} \leq \nu(\mathbf{n})$$

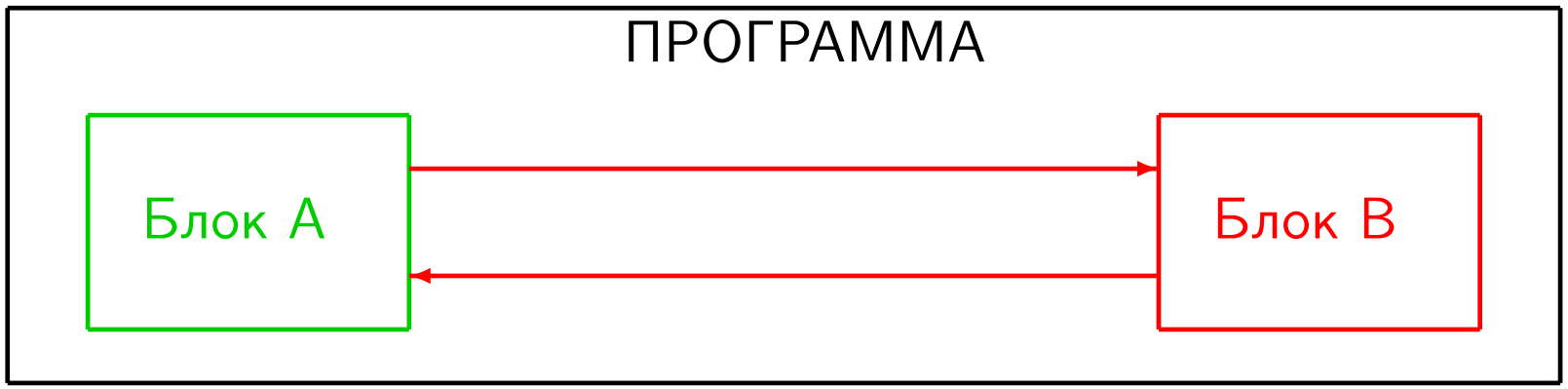
верно для любой пары констант $c_0 \in \{0,1\}^n$ и $c \in_R \{0,1\}^n$.

СТОЙКОСТЬ ОБФУСКАЦИИ

Стойкая обфускация, скрывающая константу,

- невозможна , если $\mathbf{M_x}$ — это универсальная машина Тьюринга;
- возможна , если $\mathbf{M_x} = E(key(\mathbf{x}), m)$ — это программа шифрования стойкой криптосистемы с открытым ключом $key(\mathbf{x})$ и секретным ключом $\mathbf{x}$.

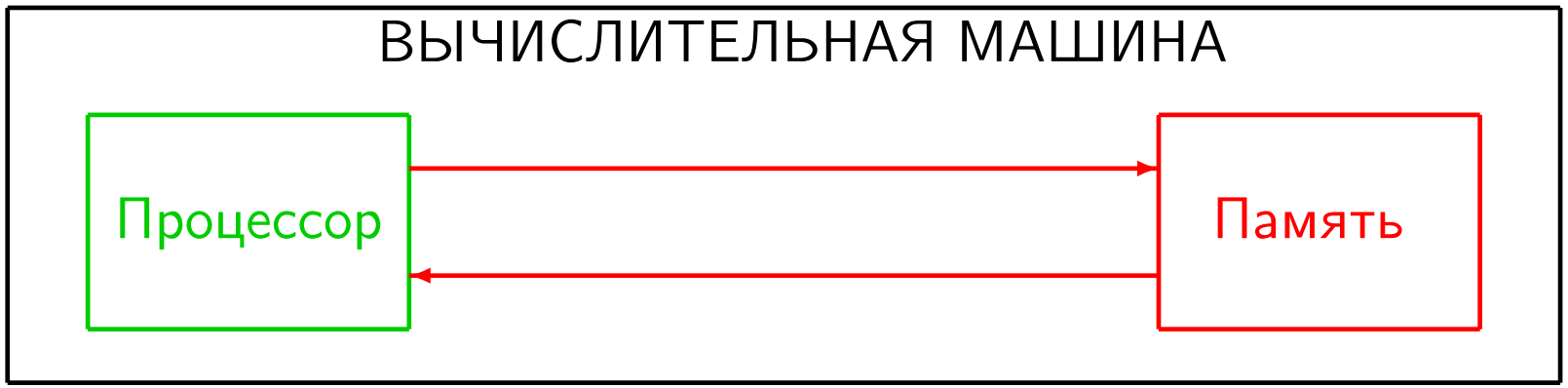
ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ



Блок A : вычисляется на защищенном устройстве небольшой вычислительной мощности (смарт-карта, процессор с малым объемом памяти и др.)

Блок B : вычисляется на открытом высокопроизводительном вычислительном устройстве.

ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ

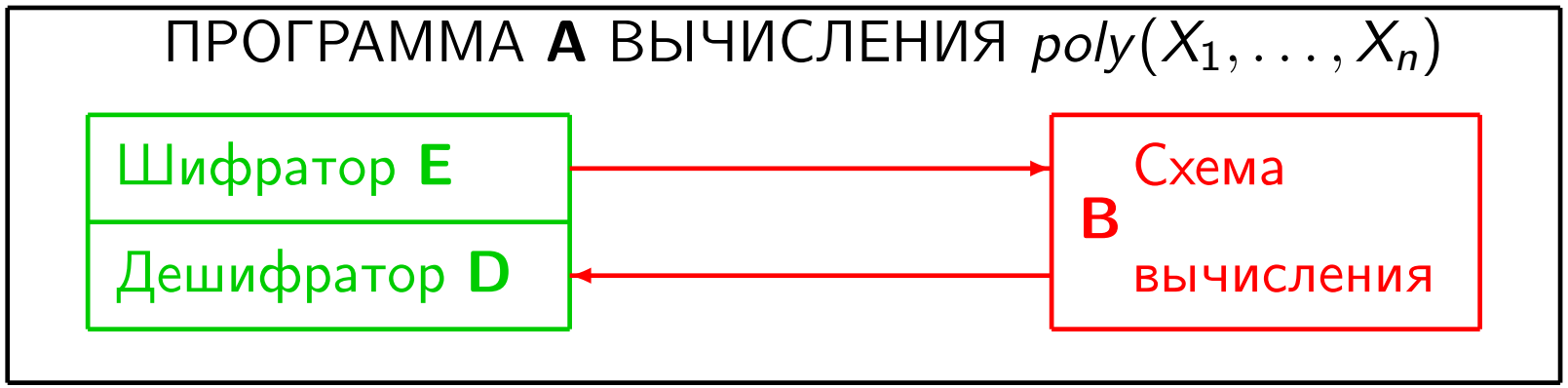


Если существуют односторонние функции, то любую программу π можно преобразовать в такую эквивалентную программу $\mathcal{O}(\pi)$, для которой:

1. $Time(\mathcal{O}(\pi)) \leq Time(\pi) \log^3(Time(\pi))$;
2. При выполнении программы $\mathcal{O}(\pi)$ на вычислительном устройстве с закрытым процессором и открытой памятью никакой противник, ограниченный полиномиальным временем, не способен распознать программу $\mathcal{O}(\pi)$ по последовательности ее обращений к памяти.

R. Ostrovsky, 1990.

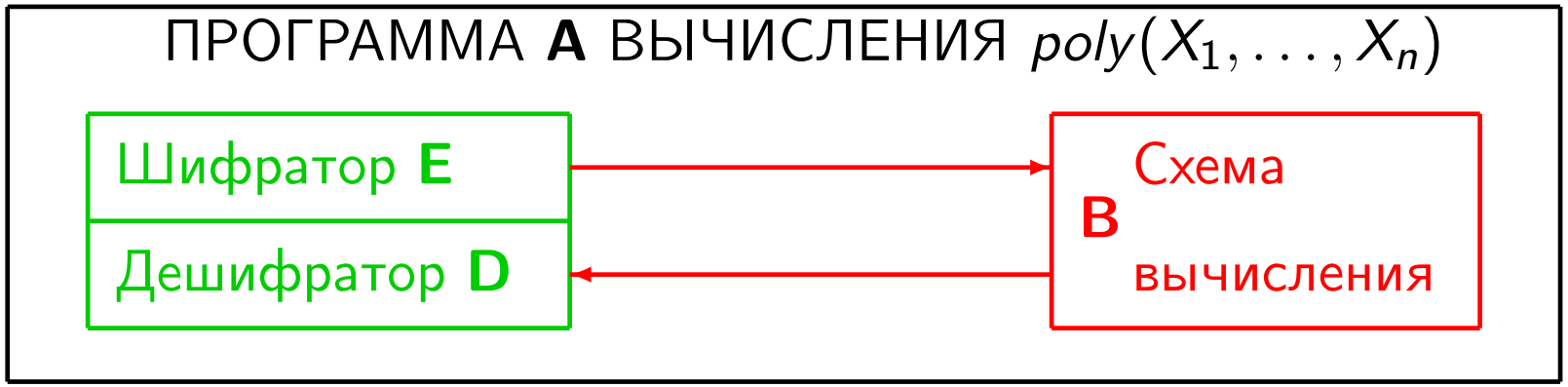
ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ



Для заданной программы **A** вычисления алгебраических выражений (алгебраической схемы) построить такие алгоритмы **E** (шифратор), **D** (дешифратор) и программу **B**, что

- **B** по эффективности незначительно отличается от **A**;
- алгоритмы **E** и **D** просты, эффективны и не зависят от **A**;
- $\mathbf{A}(x) = \mathbf{D}(\mathbf{B}(\mathbf{E}(x)))$;
- не зная алгоритмов **E** и **D**, из значения $\mathbf{E}(x)$ нельзя эффективно извлечь никакой информации о входе x;
- не зная алгоритмов **E** и **D**, из программы **B** нельзя эффективно извлечь никакой информации о параметрах (коэффициентах) программы **B**.

ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ



Теорема. [Sander, Tchudin, 1998; Шокуров А.В., 2004.]

Существует преобразование $\mathcal{O}$ ациклических программ, реализующих алгебраические вычисления, в системы вычислений над зашифрованными данными со следующими свойствами:

1. преобразование программы **A** в схему вычислений $(\mathbf{E}, \mathbf{D}, \mathbf{B})$ эффективно;
2. $size(\mathbf{B}) = poly(size(\mathbf{A}))$;
3. нижняя оценка стойкости схемы экспоненциальна .

ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ

Обфускацией с открытым ключом для класса программ $\mathcal{C}$ называется такой вероятностный алгоритм $\mathcal{O}$

$$\mathcal{O} \ : \ \mathcal{C} \ \rightarrow (\mathcal{F}, \mathcal{D}),$$

удовлетворяет следующим требованиям:

1. (функциональность) для любой программы $\pi \in \mathcal{C}$
$$\mathcal{O} = (\mathcal{F}, \mathcal{D})(\pi) \ \Rightarrow \ \pi(x) = D(F(x));$$
2. (полиномиальное замедление)
3. (стойкость) для любой PPT A (противника) и для любой пары программ $\pi_0, \pi_1 \in \mathcal{C}$ выполняется неравенство
$$|(\Pr[A(F_r) = r] - 1/2| \leq \nu(size(\pi_0) + size(\pi_1)),$$
где вероятность вычисляется по следующим переменным: случайные параметры, используемые для построения $\mathcal{O}(\pi_0) = (F_0, D_0)$, $\mathcal{O}(\pi_1) = (F_1, D_1)$, случайный бит $r \in_R \{0, 1\}$.

ОБФУСКАЦИЯ С РАЗДЕЛЕНИЕМ

Теорема. [Ostrovsky R., Skeith W., 2005.]

Существует стойкая обфускация с открытым ключом для программ выбора текстов из потока данных по заданному набору ключевых слов.

ПРОГРАММЫ, НЕПРОНИЦАЕМЫЕ ДЛЯ АНАЛИЗА

Поведение $Comp(\pi, I)$ (операционная семантика) программы π может быть определено в различных интерпретациях I.

Программы π_1 и π_2 считаются **эквивалентными** в интерпретации I, если они имеют одинаковое поведение:

$$Comp(\pi_1, I) = Comp(\pi_2, I).$$

ПРОГРАММЫ, НЕПРОНИЦАЕМЫЕ ДЛЯ АНАЛИЗА

Поведение $Comp(\pi, I)$ (операционная семантика) программы π может быть определено в различных интерпретациях I.

Программы π_1 и π_2 считаются **эквивалентными** в интерпретации I, если они имеют одинаковое поведение:

$$Comp(\pi_1, I) = Comp(\pi_2, I).$$

Семантическое свойство $\mathcal{P}$ — отображение

$$\mathcal{P} \ : \ Prog \times Int \ \rightarrow \ L,$$

где L — решетка с максимальным элементом $\top$ (полная неопределенность).

Программа π называется **$\mathcal{P}$-непроницаемой** в интерпрертации I, если выполняется равенство:

$$\mathcal{P}(\pi, I) = \top.$$

ПРОГРАММЫ, НЕПРОНИЦАЕМЫЕ ДЛЯ АНАЛИЗА

Пусть заданы абстрактные интерпретации программ I_1 и I_2 и семантическое свойство $\mathcal{P}$.

Алгоритм $\mathcal{O}$

$$\mathcal{O} \ : \ Prog \to Prog$$

называется (I_1, I_2)-полной обфускацией свойства $\mathcal{P}$, если для любой программы π выполняются следующие условия:

1. программы π и $\mathcal{O}(\pi)$ эквивалентны в интерпретации I_1;
2. программа $\mathcal{O}(\pi)$ является $\mathcal{P}$-непроницаемой в интерпретации I_2.

Теорема [Захаров В.А., Иванов К.С., 2004]

Существуют примеры интерпретаций I_1, I_2 и семантических свойств $\mathcal{P}$, для которых существуют (I_1, I_2)-полная обфускация свойства $\mathcal{P}$.

СПАСИБО ЗА ВНИМАНИЕ.

ВАШИ ВОПРОСЫ?