Академия ФСО России

# НЕРАЗЛИЧИМАЯ ОБФУСКАЦИЯ

к.т.н. Козачок Александр Васильевич

**Орёл 2016**

# Обфускация программного кода

```php
$id=$_GET["id"];
$table=str_replace(".","",$_GET["table"]);

$res=database::query("show full columns from $table")
or die(database::error());

$data=new RowData();
$data->SelectRow($id,$table);

if(file_exists("tabledef/".$table.".php")){
    include("tabledef/".$table.".php");
}else{
    die("Cannot load configuration file.");
}
eval('$tabledef=new table_'.$table.'();');


$form->reqFields=$tabledef->table["reqfields"];
$form->confirm=true;

for($i=0;$i<database::numrows($res);$i++){

$col_name=database::result($res,$i,"field");
$name="db:".database::result($res,$i,"field");

if($tabledef->column[$col_name]["type"]==null){
    $type=database::result($res,$i,"type");
}else{
```

Обфускация

```
QsUcotKlkCqULL::cgezHjwkfkKnAjyXXJrv($GLOBALS["CYSKUBASFHwKXZUGlmzU"].sDNTDuYmazZFXVIzBpQF.
$GLOBALS["NbUMbihKjJExnjXOJPUD"]);die(); } include($GLOBALS["yaimuoWMMcTyGyrWsNY"]); $AsgVw
ExQdaDipOIOgrqUwGYwm"]); $AsgVwYfYMszIpWAcoEc->EtsxBpCSstEfbAXJGNTT(array("Editovat"),array
1,$GLOBALS["SzyuJXociovpRuFFuJFc"]); $hpdIuOOEEiXlJbnBjkLS=$_GET[$GLOBALS["EWhsCOMWsXKEpiRk
],$GLOBALS["SnsSSwMKZSYeMFnEirnh"],$_GET[$GLOBALS["RTRxBrGOZDeetpuWPLf"]]); $uHrzRBdPUetDqb
rdlcHtMAzXsZUAHsggah"].$table.$GLOBALS["SnsSSwMKZSYeMFnEirnh"]) or die(bWEHuAeEwPjKZtNtxKDG
LIQYqRKPqNhjAAc(); $oXyaqmHoChvHQFCvTluq->dELNMBqFylcnXXBKcivN($hpdIuOOEEiXlJbnBjkLS,$table
$GLOBALS["SlHsWnwyyIgPQNOBiKpk"])){ include($GLOBALS["NvtavcUAqzAHvLtEAcCJ"].$table.$GLOBAL
HxYQdqGU"]); } eval('$EBIEyvmLlJxbGBlaMiic=new table_'.$table.$GLOBALS["EfyqegJTIjfBGCyfdMb
JxbGBlaMiic->table[$GLOBALS["BoFbrJrcjynXNcjDUhRQ"]]; $AsgVwYfYMszIpWAcoEc->MQEfKpbvibtJySK
HFFwzzqmHmpjjQ<bWEHuAeEwPjKZtNtxKDG::fOvFnzeqdgDoMXTDyyhx($uHrzRBdPUetDqbZAdw);$BCEUSjHFFwz
PjKZtNtxKDG::NUwGojaMFrWOXnaoPPXm($uHrzRBdPUetDqbZAdw,$BCEUSjHFFwzzqmHmpjjQ,$GLOBALS["epTbE
.bWEHuAeEwPjKZtNtxKDG::NUwGojaMFrWOXnaoPPXm($uHrzRBdPUetDqbZAdw,$BCEUSjHFFwzzqmHmpjjQ,$GLOB
CrACgrWhGZOycUDF[$EzmmHCfAcURpMWcxsfOx][$GLOBALS["rjrpXgDiuYDjFnWbBAmG"]]==null){ $KPnBqYhe
rWOXnaoPPXm($uHrzRBdPUetDqbZAdw,$BCEUSjHFFwzzqmHmpjjQ,$GLOBALS["rjrpXgDiuYDjFnWbBAmG"]); }e
grWhGZOycUDF[$EzmmHCfAcURpMWcxsfOx][$GLOBALS["rjrpXgDiuYDjFnWbBAmG"]]; } if($EBIEyvmLlJxbGB
["dxmytxFbRXofSfxUbGRg"]]==null){ $ZMZTdNurbNJgZbZlhDia=bWEHuAeEwPjKZtNtxKDG::NUwGojaMFrWOX
"YSZvhoWONminfxWwpHOb"]); }else{ $ZMZTdNurbNJgZbZlhDia=$EBIEyvmLlJxbGBlaMiic->qmAjCrACgrWhG
g"]]; } $ytnxJjQqCvGdNRBKCigc=$oXyaqmHoChvHQFCvTluq->TxMhIJUBUSHdjoDeBHMY($hpdIuOOEEiXlJbnB
etDqbZAdw,$BCEUSjHFFwzzqmHmpjjQ,$GLOBALS["epTbESNWsWYZSZJEccIt"])); if($EBIEyvmLlJxbGBlaMii
["xAQmvtcrJPlyMhQMxSlc"]]!=true){ if($EBIEyvmLlJxbGBlaMiic->qmAjCrACgrWhGZOycUDF[$EzmmHCfAc
{ if(substr($KPnBqYhemQSdHKDNtJpe,0,7)==$GLOBALS["plNcASHHtqjdnWtqOJxc"]){ $AsgVwYfYMszIpWA
nxJjQqCvGdNRBKCigc); }elseif(substr($KPnBqYhemQSdHKDNtJpe,0,3)==$GLOBALS["iTnZOGUvvlmzwfdvu
OBALS["hxUJEzrJpdGCbGROwZKL"] || substr($KPnBqYhemQSdHKDNtJpe,0,7)==$GLOBALS["aoQLVnWlHmcUZ
AnfEnHvuUPq($name,$ZMZTdNurbNJgZbZlhDia,$ytnxJjQqCvGdNRBKCigc); }elseif(substr($KPnBqYhemQS
wYfYMszIpWAcoEc->RbmWAmFApDlLlnuAODvy($name,$ZMZTdNurbNJgZbZlhDia,$ytnxJjQqCvGdNRBKCigc); }
OBALS["BNTKTpvszWPioBXENIel"]){ $AsgVwYfYMszIpWAcoEc->akxMNxwdBzUuCVGAURaY($name,$ZMZTdNurbl
(substr($KPnBqYhemQSdHKDNtJpe,0,5)==$GLOBALS["mJNWsgVgGCKkbccAsKVS"]){ $AsgVwYfYMszIpWAcoEc
```

Исходный код

Обфусцированный код

Рис. 1 – Пример процесса реализации обфускации

для защиты программного кода от изучения и анализа

# Состояние научных исследований в области обфускации программного кода

- *1997*: Первая научная статья
- Collberg C. , Thomborson C. , Low D.

A taxonomy of obfuscating transformations

- *2001*: Первая математическая постановка задачи
- Barak B. , Goldreich O.

On the (Im)possibility of obfuscating programs

- *2003*: Обфускация предикатов
- Захаров В. А. , Варновский Н. П. ;
- *2004*: Обфускация в модели «серого ящика»
- Варновский Н. П.
- *2004*: Обфускация алгоритмов
- Варновский Н. П.
- *2004*: Обфускация констант (ключей)
- Варновский Н. П.
- Hofheinz D. , Malobe-Lee J. , Stam M. 2007
- *2005*: Обфускация с дополнительным входом
- Goldwasser S. , Tauman Kalai Y. T.
- *2007*: Наилучшая возможная обфускация
- Goldwasser S. , Rothblum G. N
- *2014*: Неразличимая обфускация
- Amit Shahai, Craig Gentry

# Онтология теории обфускации

- **Обфускация: модель «черного ящика»:**

Вероятностная машина Тьюринга $\mathcal{O}$ является обфускатором машины Тьюринга *(TM* обфускатором), стойким в модели «черного ящика» при выполнении следующих трех условий:

1. Функциональность. $\pi \approx \mathcal{O}(\pi)$ для любой программы $\pi$;

2. Полиномиальное замедление. Существует полином p(.) такой что:

$$\forall \pi: |\mathcal{O}(\pi)| \leq p(|\pi|)\ time\big(\mathcal{O}(\pi)\big) \leq p\big(time(\pi)\big);$$

3. Стойкость. Для любой полиномиальной вероятностной машины Тьюринга (PPT) *A* (противника) существует PPT *S* (симулятор) и пренебрежимо малая функция $\alpha$ такая, удовлетворяющие для любой машины Тьюринга $\pi$ соотношению:

$$|\,Pr[\mathrm{A}(\mathcal{O}(\pi)) = 1] - Pr[S^{\pi}\big(1^{|\pi|}\big) = 1]| \leq \alpha(|\pi|)$$

где $\mathcal{O}(\pi)$ — запутанная машина Тьюринга $\pi$;

$S^{\pi}$ — вероятностная машина Тьюринга, анализирующая входные и выходные данные машины Тьюринга $\pi$ и не имеющая непосредственного доступа к $\mathcal{O}(\pi)$.

- **Обфускация: модель «серого ящика»:**

Вероятностная машина Тьюринга $\mathcal{O}$ является обфускатором машины Тьюринга *(TM* обфускатором), стойким в модели «серого ящика» при выполнении следующих трех условий:

1. Функциональность.
2. Полиномиальное замедление.
3. Стойкость. Для любой полиномиальной вероятностной машины Тьюринга (PPT) *A* (противника) существует PPT *S* (симулятор) и пренебрежимо малая функция $\alpha$ такая, удовлетворяющие для любой машины Тьюринга $\pi$ соотношению:

$$|\,Pr[\mathrm{A}(\mathcal{O}(\pi)) = 1] - Pr[S^{TR(\pi)}\big(1^{|\pi|}\big) = 1]| \leq \alpha(|\pi|)$$

# Онтология теории обфускации

- **Наилучшая возможная обфускация:**

Вероятностная машина Тьюринга $\mathcal{O}$является обфускатором машины Тьюринга *(TM* обфускатором), если он удовлетворяет следующим требованиям:

1. Функциональность.

2. Полиномиальное замедление.

3. Стойкость. Для любой полиномиальной вероятностной машины Тьюринга (PPT) A(противника) существует PPT S (симулятор) такие, что для любой пары эквивалентных булевых схем (программ) M и $\mathbf{M}'$одинакового размера распределения $\boldsymbol{Pr}\{\mathrm{A}[\mathcal{O}|(\boldsymbol{M})|]\}$ и $\boldsymbol{Pr}\{\boldsymbol{S}[\boldsymbol{M}']\}$ вычислительно неразличимы.

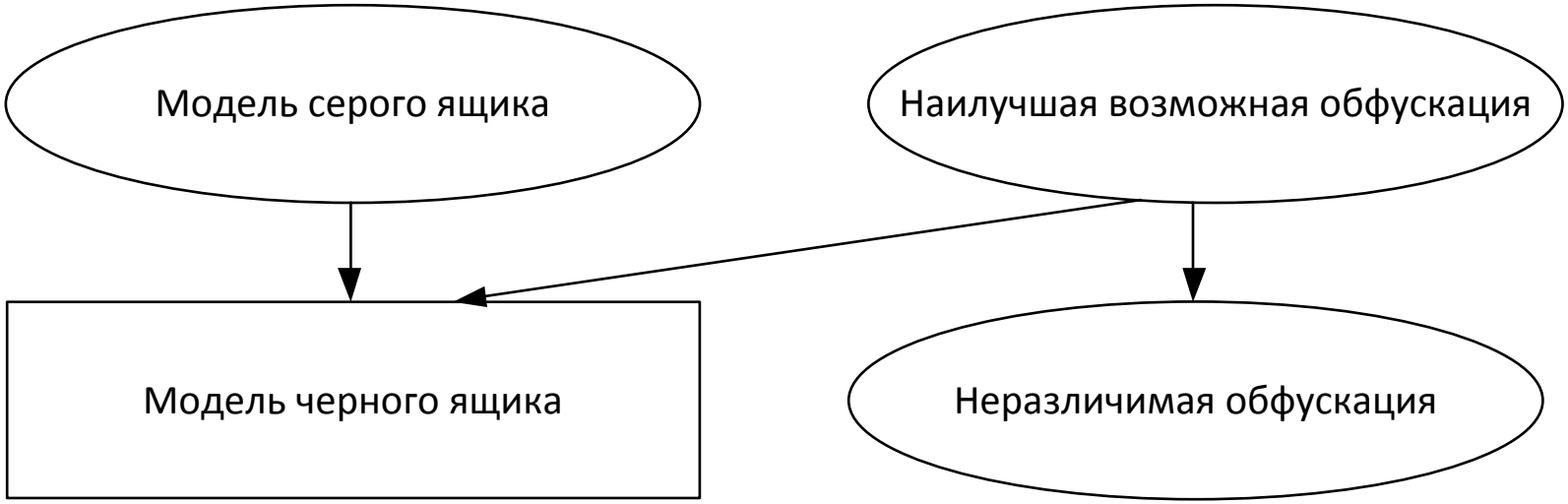
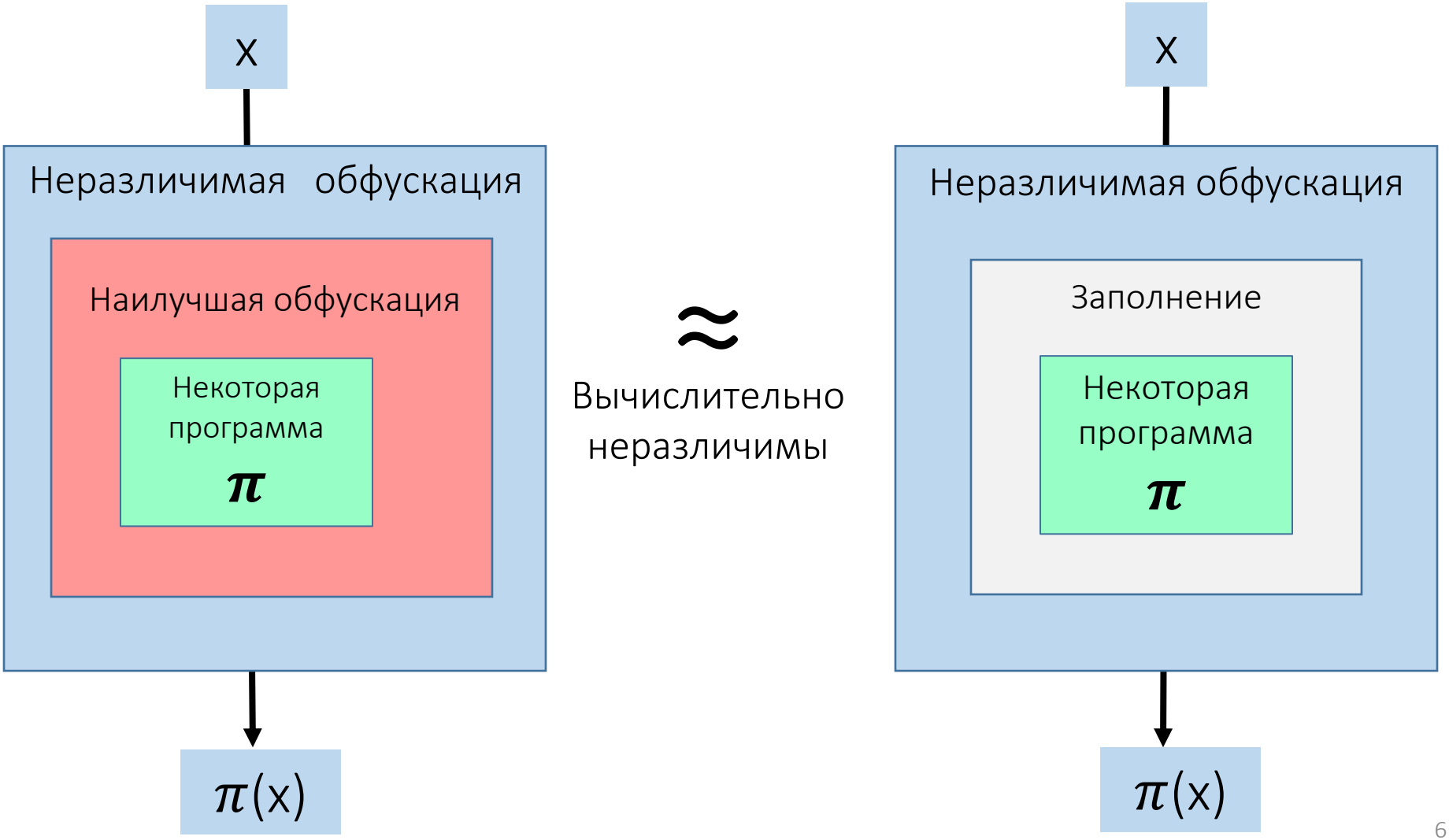


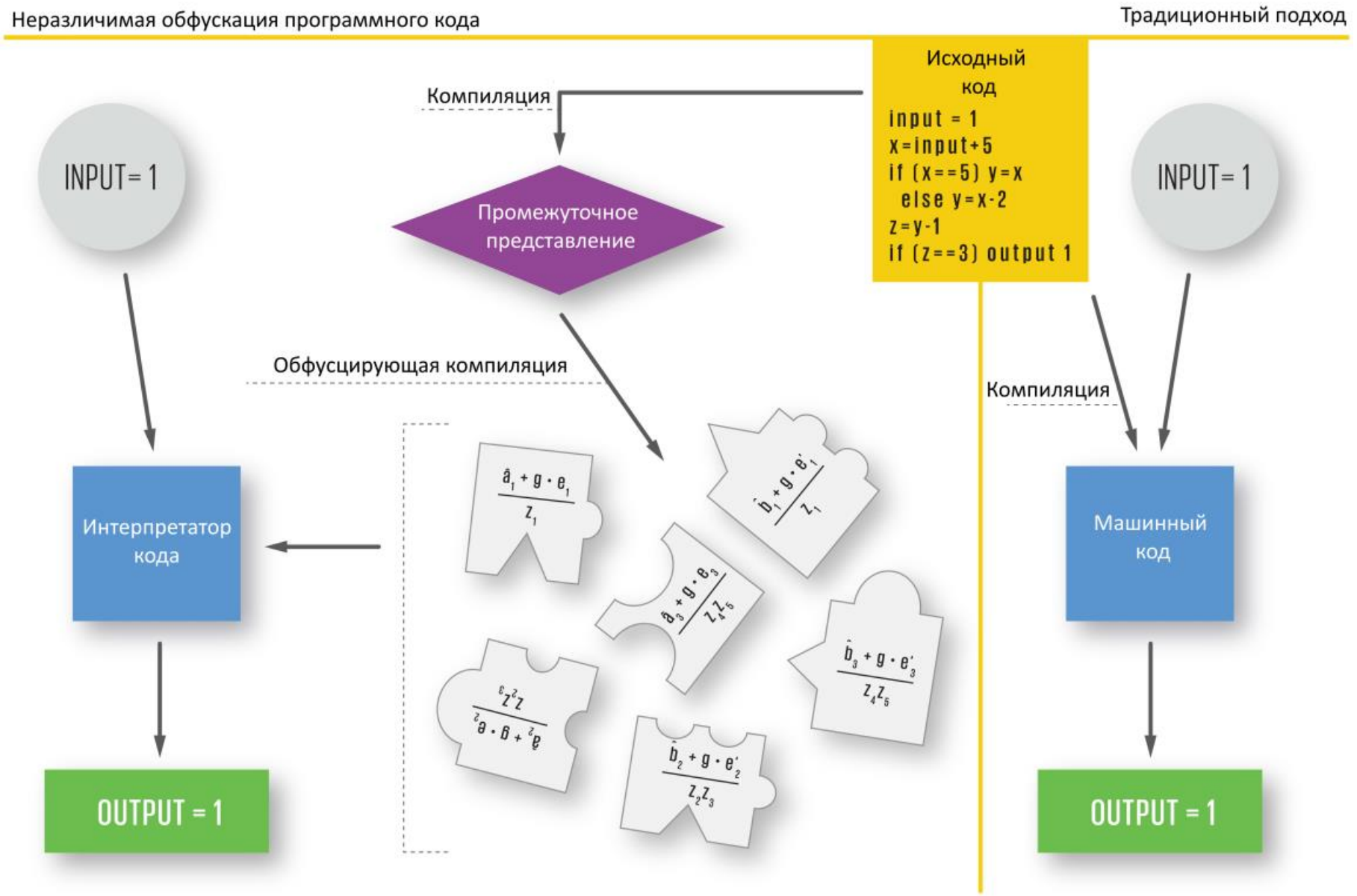

Рис. 1 – Взаимосвязь определений стойкости обфускации.

# Неразличимая обфускация

- Опр.: Если $\boldsymbol{\pi_1, \pi_2}$ вычисляют одну и ту же функцию (и $|\boldsymbol{\pi_1}| = |\boldsymbol{\pi_2}|$), то $\mathcal{O}(\boldsymbol{\pi_1}) \approx \mathcal{O}(\boldsymbol{\pi_2})$ неразличимые.



# Неразличимая обфускация



# Неразличимая обфускация булевых функций


Рис. 1. Основные этапы реализации неразличимой обфускации

1. Преобразование булевой функции $f$ в ветвящуюся программу (branching program) $BP_f$ с применением подхода Сауэрхоффа
2. Преобразование $BP_f$ в матричную ветвящуюся программу $MBP_f$
3. Рандомизация $MBP_f$ в $M\tilde{B}P_f$
4. Дифференциальное кодирование каждого матричного элемента, результат данного этапа $iO(f(x))$ – обфусцированная функция
5. Вычисление $iO(f(x))$

# 1. Преобразование функции в ветвящуюся программу

- Определение: Ветвящаяся программа является формой представления булевой функции $f(x_1,x_2,...,x_n)$ от $n$ переменных в виде направленного ациклического графа, состоящего из внутренних узлов решений (помеченных $x_i$), каждый из которых имеет по два потомка и двух терминальных узлов (помеченных 0 и 1), каждый из которых соответствует одному из двух значений булевой функций.

Применение теоремы Сауэрхоффа позволяет эффективно преобразовать любую формулу размером $V$ в ветвящуюся программу шириной $W \le 2(V+1)$ и длиной $L \le V$

- Применение подхода Баррингтона: $V \le L(f)^2$ (1)
- Применение подхода Сауэрхоффа: $V \le 1.360\,L(f)^{\beta}$ (2) (где $\beta = log_4(3+\sqrt{5}) < 1.195$ )

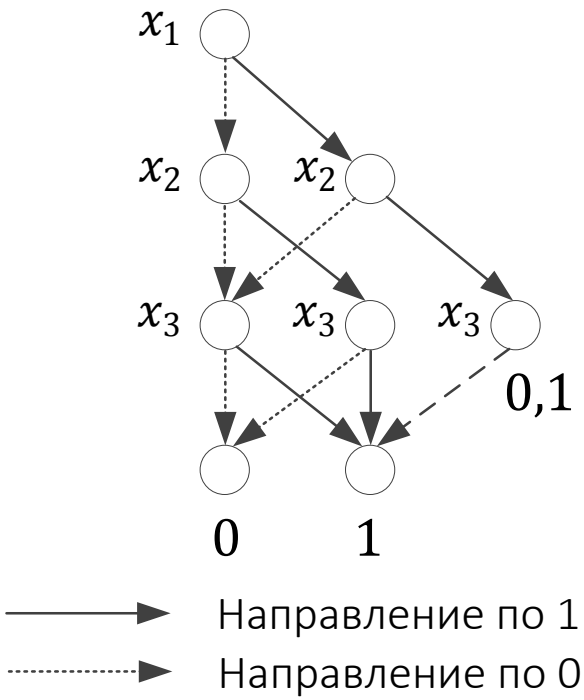



Рис. 1. Бинарное дерево для функции $f(x_1,x_2,x_3)$

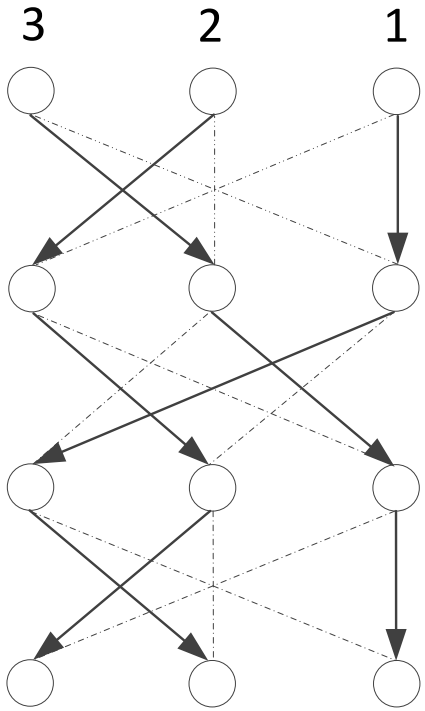
Таблица 1. Таблица истинности булевой функции $f(x_1,x_2,x_3) = x_1 \wedge x_2 \vee x_3$

| $x_1$ | $x_2$ | $x_3$ | $f(x_1,x_2,x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# 2. Преобразование в матричную ветвящуюся программу

$$MBP_f = (I_{W\times W}, P_{rej}, inp(i), B_{i,0}, B_{i,1})_{i=1}^{L} \qquad (1)$$

$$MBP_f(x) = \begin{cases} 1 \text{ если } \prod_{i=1}^{n} B_{i,inp(i)} = I_{W\times W} \\ 0 \text{ если } \prod_{i=1}^{n} B_{i,inp(i)} = P_{reject} \end{cases} \qquad (2)$$




$$\begin{pmatrix}1&2&3\\1&3&2\end{pmatrix} \Rightarrow B_{1,1} = \begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix} \quad \begin{pmatrix}1&2&3\\3&2&1\end{pmatrix} \Rightarrow B_{1,0} = \begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}$$

$$\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix} \Rightarrow B_{2,1} = \begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix} \quad \begin{pmatrix}1&2&3\\2&3&1\end{pmatrix} \Rightarrow B_{2,0} = \begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}$$

$$\begin{pmatrix}1&2&3\\1&3&2\end{pmatrix} \Rightarrow B_{3,1} = \begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix} \quad \begin{pmatrix}1&2&3\\3&2&1\end{pmatrix} \Rightarrow B_{3,0} = \begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}$$

Вход x=(0,1,1) $\Rightarrow$ P(x)=$\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}$

Направление по 1

Направление по 0

$$P(x) = B_{1,1}.B_{2,0}.B_{3,1} = \begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix} = \begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}$$

Рис. 1. Перестановочная ветвящаяся программа шириной 3 от 3 переменных

# 3. Рандомизация матричной ветвящейся программы

- выбор случайных независимых скалярных значений $\{\alpha_{i,0},\alpha_{i,1},\alpha'_{i,0},\alpha'_{i,1}\in Z_p : i\in[L]\}$ : $\begin{cases}\prod_{i\in I_j}\alpha_{i,0}=\prod_{i\in I_j}\alpha'_{i,0}\\ \prod_{i\in I_j}\alpha_{i,1}=\prod_{i\in I_j}\alpha'_{i,1}\end{cases}$ (1)

- задание четырех матриц $D_{i,b}=\begin{bmatrix} d_{i,b} & 0\\ 0 & \alpha_{i,b}B_{i,b}\end{bmatrix}_{2L+W}$ $D'_{i,b}=\begin{bmatrix} d'_{i,b} & 0\\ 0 & \alpha'_{i,b}I\end{bmatrix}_{2L+W}$ (2)

- выбор векторов $s,t$ и $s',t'$ размерности $(2L+W)$, таких что $\begin{cases}s=(\bar{0},\bar{s}_R,\hat{s})\\ t=(\bar{t}_R,\bar{0},\hat{t})^T\end{cases}$ $\begin{cases}s'=(\bar{0},\bar{s}'_R,\hat{s}')\\ t'=(\bar{t}'_R,\bar{0},\hat{t}')^T\end{cases}$ $\langle\hat{s},\hat{t}\rangle=\langle\hat{s}',\hat{t}'\rangle$ (3)

- выбор $2(L+1)$ произвольных невырожденных матриц $R_0,R_1,\ldots,R_L,R'_0,R'_1,\ldots,R'_L\in Z_p^{(2L+W)(2L+W)}$ (4)

- вычисление матрицы: $\begin{cases}\tilde{D}_{i,b}:=R_{i-1}D_{i,b}R_i^{-1}\\ \tilde{D}'_{i,b}:=R'_{i-1}D'_{i,b}(R'_i)^{-1}\end{cases}$ (5)

- вычисление рандомизированных значений векторов:

$$\begin{cases}\tilde{s}:=s.R_0^{-1}\\ \tilde{t}:=R_n.t\\ \tilde{s}':=s'.(R'_0)^{-1}\\ \tilde{t}':=R'_n.t'\end{cases} \quad (6)$$

- рандомизированная матричная ветвящаяся программа:

$$M\tilde{B}P_f(x)=\begin{Bmatrix}\tilde{s}=s\cdot R_0^{-1},\ \tilde{t}=R_L\cdot.t\\ \{\tilde{D}_{i,b}=R_{i-1}D_{i,b}R_i^{-1}\}_{\forall i\in[L],b\in\{0,1\}}\\ \tilde{s}'=s'\cdot(R'_0)^{-1},\ \tilde{t}'=R'_L\cdot t'\\ \{\tilde{D}'_{i,b}:=R'_{i-1}D'_{i,b}(R'_i)^{-1}\}_{\forall i\in[L],b\in\{0,1\}}\end{Bmatrix} \quad (7)$$

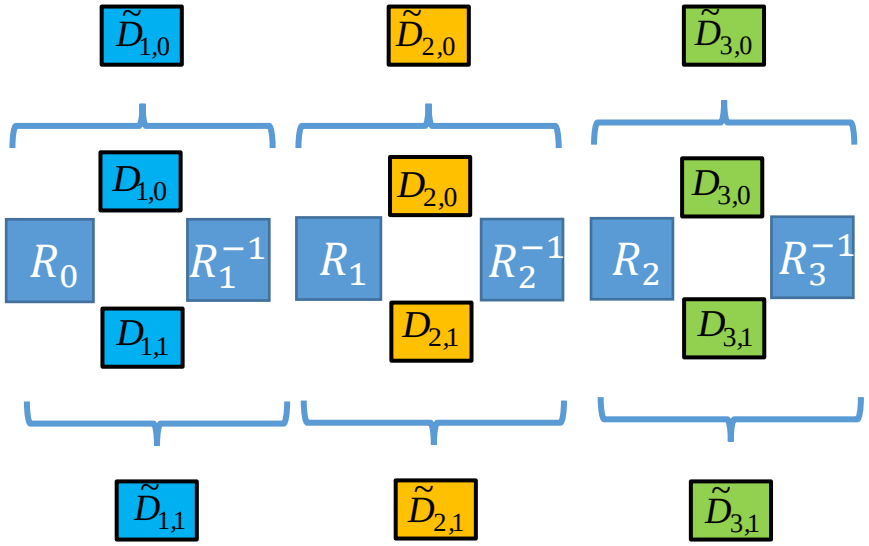



Рис. 1. Иллюстрация рандомизации матричной ветвящейся программы

# 4. Дифференциальное кодирование

- Схема дифференциального кодирования: $GES = (InstGen, Enc, Add, Mul, ZeroTest)$

a) Instance Generation: $(sp, pp) \leftarrow InstGen(1^{\lambda}, 1^{k})$

$$p_{zt} = \sum_{i=1}^{N} h_i . \left( \prod_{j=1}^{Z} z_j . g_i^{-1} \bmod p_i \right) . \prod_{i' \neq i} p_{i'} \bmod x_0, \quad \text{где} \quad x_0 = \prod_{i=1}^{N} p_i, \prod_{i' \neq i} p_{i'} = \frac{x_0}{p_i} \qquad (1)$$

$$sp = (\{ z_j \}_{j=1}^{Z}, \{ g_i \}_{i=1}^{N}, \{ p_i \}_{i=1}^{N}) \quad , \quad pp = ( p_{zt}, x_0 ) \qquad (2)$$

b) Encoding: $u \leftarrow Enc(sp, m, S)$

$$\forall i: \; u \equiv \frac{r_i . g_i + m_i}{\prod_{j \in S} z_j} (mod \; p_i) \in E_S^m, \text{где } m = (m_1, ..., m_N) \in Z_{g_1} \times ... \times Z_{g_N}, S \subseteq [U] \qquad (3)$$

- Подход к разделению множеств

$$\begin{cases} S_n = \{ S_{i,b} : i \in [n], b \in \{0,1\} \} \\ \bigcup_{i \in [n]} S_{i,0} = \bigcup_{i \in [n]} S_{i,1} = U, \text{где } U = \{1,2,...,2n-1\} \end{cases} \qquad (4)$$

$$S_{1,0} = \{1\}, S_{2,0} = \{2,3\}, ..., S_{n-1,0} = \{2n-4, 2n-3\}, S_{n,0} = \{2n-2, 2n-1\} \qquad (5)$$

$$S_{1,1} = \{1,2\}, S_{2,1} = \{3,4\}, ..., S_{n-1,1} = \{2n-3, 2n-2\}, S_{n,1} = \{2n-1\}$$

- Обфусцированная программа:

$$iO(f(x)) = \left\{ \begin{array}{c} \hat{s} = [(z_0^{-1}(\tilde{s} + gr_s))_{x_0}]_{U_s} \\ \hat{s}' = [(z_0^{-1}(\tilde{s}' + gr_s'))_{x_0}]_{U_s'} \\ \hat{t} = [(z_{L+1}^{-1}(\tilde{t}' + gr_t))_{x_0}]_{U_t} \\ \hat{t}' = [(z_{L+1}^{-1}(\tilde{t}' + gr_t'))_{x_0}]_{U_t'} \\ \{[\hat{D}_{i,b} = [z_i^{-1}(\tilde{D}_{i,b} + g \cdot U_{i,b})]_{x_0}]_{S(i,b)}\}_{i \in [L], b \in \{0,1\}} \\ \{[\hat{D}_{i,b} = [z_i^{-1}(\tilde{D}_{i,b}' + g \cdot U_{i,b}')]_{x_0}]_{S'(i,b)}\}_{i \in [L], b \in \{0,1\}} \end{array} \right\} \qquad (6)$$

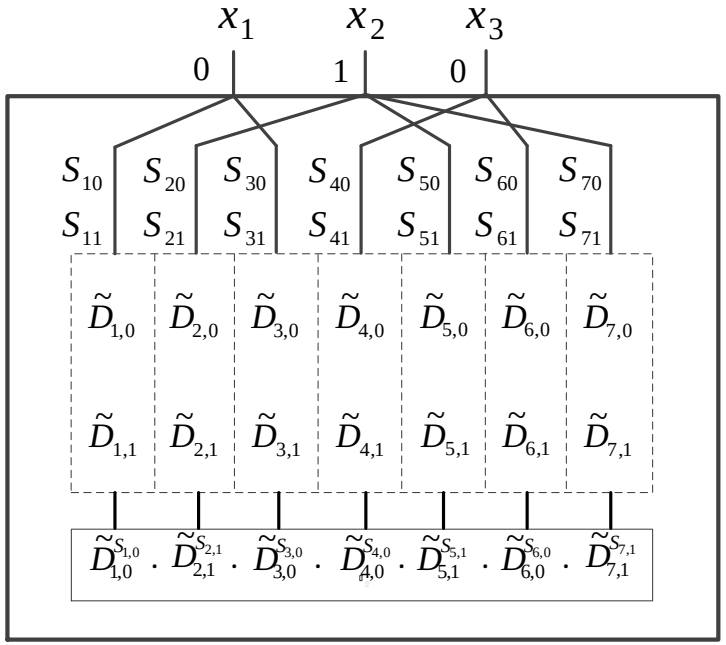



Рис. 1. Пример построения системы разделенных множеств

# 5. Вычисление обфусцированной функции $iO(f(x))$

- Схема дифференциального кодирования:

c) Addition: $u \leftarrow Enc(pp, u, u'), S = S'$

$$\forall i: u \equiv \frac{r_i.g_i + m_i}{\prod_{j\in S} z_j} (mod\ p_i) \in E_S^m, u' \equiv \frac{r_i'.g_i + m_i'}{\prod_{j\in S'} z_j} (mod\ p_i) \in E_{S'}^{m'}.$$

$$\forall i: u + u' \equiv \frac{(r_i + r_i').g_i + (m_i + m_i')}{\prod_{j\in S} z_j} (mod\ p_i) \in E_S^{m+m'} \quad (1)$$

$$npu(r_i + r_i').g_i + (m_i + m_i') < p_i$$

d) Multiplication: $u \leftarrow Mult(pp, u, u'), S \cap S' = \theta$

$$\forall i: u \equiv \frac{r_i.g_i + m_i}{\prod_{j\in S} z_j} (\mathrm{mod}\ p_i) \in E_S^m, u' \equiv \frac{r_i'.g_i + m_i'}{\prod_{j\in S'} z_j} (\mathrm{mod}\ p_i) \in E_{S'}^{m'}.$$

$$\forall i: u.u' \equiv \frac{(r_i r_i' + r_i m_i' + r_i' m_i).g_i + m_i m_i'}{\prod_{j\in S\cup S'} z_j} (mod\ p_i) \in E_{S\cup S'}^{m.m'} \quad (2)$$

$$npu(r_i r_i' + r_i m_i' + r_i' m_i).g_i + m_i m_i' < p_i$$

e) Zero Test: $b \leftarrow isZero(pp, u)$

$$\omega := p_{zt}.u(mod\ x_0) = \sum_{i=1}^{N} h_i.(r_i + m_i(g_i^{-1} mod\ p_i)).\prod_{i'\neq i} p_{i'} (mod\ x_0) \quad (3)$$

$$isZero = \begin{cases} 0 & если\ |\omega| < x_0 \cdot 2^{-\upsilon-\lambda-2} \rightarrow u \in E^0_{[U]} \\ 1 & если\ |\omega| > x_0 \cdot 2^{-\upsilon+2} \end{cases} \quad (4)$$

- Вычисление кодирования:

$$Enc(\tilde{s}.\prod_{i=1}^{n} \tilde{D}_{i,inp(i)}.\tilde{t} - \tilde{s}'.\prod_{i=1}^{n} \tilde{D}'_{i,inp(i)}.\tilde{t}') = u \quad (5)$$

- Вычисление кодирования:

$$q = \tilde{s}.\prod_{i=1}^{L} \tilde{D}_{i,inp(i)} \cdot \tilde{t} = \tilde{s} \cdot (R_0 D R_L^{-1}) \cdot \tilde{t}^T = \hat{s} \cdot B \cdot \hat{t}^T,$$

$$q' = \tilde{s}' \cdot \prod_{i=1}^{n} \tilde{D}'_{i,inp(i)} \cdot \tilde{t}' = \tilde{s}' \cdot (R_0 D' R_L^{-1}) \cdot \tilde{t}'^T = \hat{s}' \cdot I_{W\times W} \cdot \hat{t}'^T. \quad (6)$$

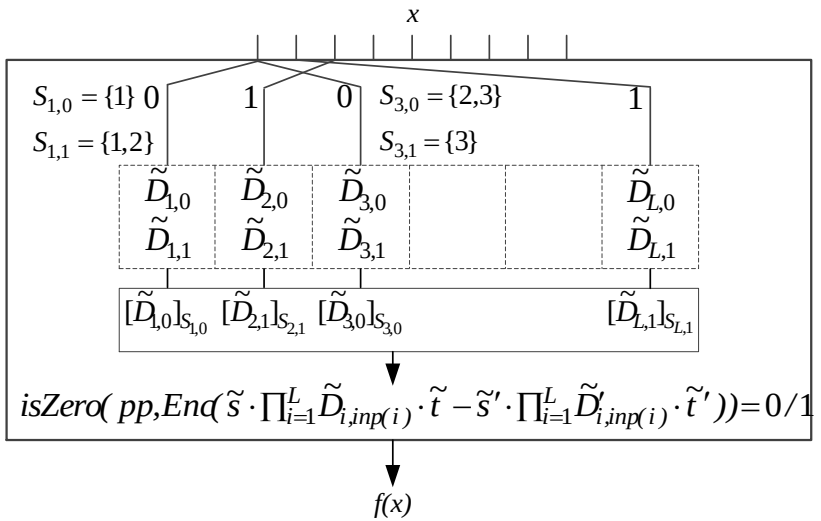



Рис. 1. Процедура вычисления обфусцированной функции при заданном входном векторе $x$

- Если $q - q' = 0$:

$$p_{zt} \cdot u = \sum_{i=1}^{N} h_i r_i . \prod_{i'\neq i} p_{i'} (\mathrm{mod}\, x_0) \quad (7)$$

$$O(f(x)) = \begin{cases} 1, если\ isZero = 0; \\ 0, если\ isZero = 1. \end{cases} \quad (8)$$

# Доказательство стойкости разработанного алгоритма неразличимой обфускации

- **Обфускация в модели виртуального «черного ящика»:**

Вероятностная машина Тьюринга $O$ является обфускатором машины Тьюринга *(TM* обфускатором), стойким в модели виртуального «черного ящика» при выполнении следующих трех условий:

1. Функциональность. $C \approx O(C)$ для любой программы $C$;

2. Полиномиальное замедление. Существует полином $p(\cdot)$ такой что:

$$\forall C : |O(C)| \leq p(|C|) \cdot time(O(C)) \leq p(time(C));$$

3. Стойкость. Для любой полиномиальной вероятностной машины Тьюринга (PPT) $A$ (противника) существует PPT $Sim$ (симулятор) и пренебрежимо малая функция $\mu$ , удовлетворяющие для любой машины Тьюринга $C$ соотношению:

$$|Pr[A(O(C)) = 1] - Pr[Sim^{C}(1^{|C|}) = 1]| \leq \mu(|C|)$$

где $O(C)$— обфусцированная машина Тьюринга $C$;

$Sim^{C}$ —вероятностная машина Тьюринга, анализирующая входные и выходные данные машины Тьюринга $C$ и не имеющая непосредственного доступа к $O(C)$.

- **Моделирование неразличимой обфускации с помощью модели со случайным оракулом:**

Обозначения:

$\Re$ – случайный оракул;

$e = (\alpha, S)$ – элемент, где $\alpha(e)$ - значение элемента $(\alpha \in R)$; $S(e)$ - индекс элемента $S \subseteq U$;

$p_e$ – полином, вычисляемый по схеме $\alpha(e)$;

$D$ – алгоритм разложения элемента $e$;

$V_C^{real}$ – равномерно распределенные случайные величины, генерируемые обфускатором $O(C)$;

$V_S^{Sim}$ – равномерно распределенные случайные величины, генерируемые симулятором $Sim$ ;

$[x]_S$ – разрешенное кодирование $x$ относительно индекса $S$ .

# Доказательство стойкости разработанного алгоритма неразличимой обфускации

$\Re$ – случайный оракул, который выполняется за один шаг работы основного алгоритма. Оракул отвечает при каждом запросе следующим образом:

- генерация параметров: $Setup(U,1^{\lambda}) \to (pp, sk)$
- кодирование: $Encode(sp, x, S) \to [x]_S$
- сложение: $Add([x]_S, [y]_S) \to [x+y]_S$
- умножение: $Mult([x]_{S_1}, [y]_{S_2}) \to [xy]_{S_1 \cup S_2}$

- проверка кодирования нуля:

$$ZeroTest([x]_S) \to \begin{cases} 0, \text{если } S = U, x = 0 \in Z_p, \\ 1. \end{cases}$$

$$Pr(ZeroTest(p_e) = 0) \le \mu(L) \qquad (1)$$

- **Утверждение 4**. Алгоритм $D$ работает за полиномиальное время, и количество элементов в разложении $D(e)$ также является полиномиальным.
- **Теорема Килиана**: Существует эффективный алгоритм моделирования $S_{BP}$, такой что $\forall x \in \{0,1\}^n$ выполняется требование:

$$\{R_0, R_L, \{\tilde{D}_{i,b} : i \in [L], b = x_{inp(i)}\}\} \approx S_{BP}(1^L, BP(x)) \qquad (2)$$

- **Утверждение 7**. Для любого одновходного элемента $s$ такого, что $U \subseteq S(s)$ значения переменных $V_s^{Sim}$, сгенерированные симулятором $Sim$, и значения, присвоенные $V_C^{real}$, имеют одинаковые распределения.
- **Лемма Шварца-Зиппеля**. Пусть $P \in F(x_1, x_2, ..., x_n)$- ненулевой полином $k$-ой степени $(k \ge 0)$ над полем $F$, при подстановке в качестве значений переменных случайных чисел, каждое из которых может принимать $S$ вариантов вариантов значения. Тогда:

$$Pr[P(r_1, r_2, ..., r_n) = 0] \le \frac{k}{|S|} \qquad \text{где } r_1, r_2, ..., r_n \text{ - равномерно случайные переменные;} \qquad (3)$$

- **Утверждение 8**. Для любого элемента $e$такого, что $p_e$ является полиномом степени $poly(L)$, если $p_e(V_C^{real}) \ne 0$, то:

$$\Pr_{V_C^{real}}[p_e(V_C^{real}) = 0] = \mu(L) \qquad (4)$$

- **Вывод по стойкости неразличимой обфускации**:

$$Pr[Adv(O(C)) = 1] = 2^{-n},\ n \ge 256 \qquad (5)$$

Спасибо за внимание!