УДК 004.056.53

А.В. Козачок, М.В. Бочков, Р.Р. Фаткиева, Л.М. Туан

АНАЛИТИЧЕСКАЯ МОДЕЛЬ ЗАЩИТЫ ФАЙЛОВ ДОКУМЕНТАЛЬНЫХ ФОРМАТОВ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Козачок А.В., Бочков М.В., Фаткиева Р.Р., Туан Л.М. **Аналитическая модель защиты файлов документальных форматов от несанкционированного доступа.**
Аннотация. В статье представлено описание метода защиты от несанкционированного доступа, основанного на применении процедуры неразличимой обфускации. Обосновано применение неразличимой обфускации для решения задачи защиты от несанкционированного доступа. Предложена математическая модель неразличимой обфускации программного кода, положенная в основу метода защиты от несанкционированного доступа к файлам документальных форматов.
Ключевые слова: обфускация, булева функция, несанкционированный доступ, защита.

Kozachok A.V., Bochkov M.V., Fatkieva R.R., Tuan L.M . **Analytical Model for Protecting Documentary File Formats from Unauthorized Access.**
Abstract. The article describes the method of documentary file formats protection from unauthorized access based on indistinguishable program code obfuscation. The application of indistinguishable obfuscation to solve the problem of unauthorized access protection is substantiated. Mathematical model of indistinguishable program code obfuscation underlying the method of documentary file formats protection from unauthorized access is proposed.
Keywords: obfuscation, Boolean function, unauthorized access, protection.

1. Введение. В последние годы особое внимание уделяется вопросам обеспечения безопасности инфокоммуникационных систем. Главным образом, эти исследования касаются формальной верификации свойств безопасности. Основной целью при этом является разработка формальной математической модели свойств безопасности в системе, а также верификация этой модели с помощью математических доказательств.

Для обеспечения безопасности инфокоммуникационных систем в течение многих лет акцент делался на обеспечение контроля доступа субъектов к объектам доступа. Невозможность полностью устранить утечку информации при этом подходе постепенно привела специалистов в области обеспечения безопасности к пониманию важности исследований информационных потоков в системе. Было предложено большое количество формальных моделей поведения системы и определения информационных потоков.

Целью проводимого исследования является построение комплекса моделей и алгоритмов процесса контролируемого разграничения доступа к файлам документальных форматов, позволяющего осуществить защиту от несанкционированного доступа к информации за счет применения неразличимой обфускации программного кода [1].

Исходя из вышеизложенного, в статье предлагается аналитическая модель защиты от несанкционированного доступа к файлам документальных форматов, отличающаяся применением процедуры неразличимой обфускации программного кода. При этом субъектами доступа в модели выступают пользователи, идентифицируемые учетными записями, а объектами являются файлы документальных форматов. Правила разграничения доступа субъектов к объектам задаются в виде матрицы полномочий, учитывающей метки конфиденциальности.

Предлагаемая модель позволяет хранить файлы документальных форматов в унифицированном виде и обеспечивает единый метод доступа к ним. Для безопасного хранения данных используется формат защищенного контейнера, в котором данные хранятся в обфусцированном виде. Контейнер представляет собой исполняемый файл, в который инкапсулируются файлы документальных форматов, обладающий рядом заданных свойств и функций, позволяющих однозначно идентифицировать пользователя, разграничивать доступ к данным, обеспечивать защиту конфиденциальности внедренного документа. Формат контейнера обеспечивает его безопасное хранение и передачу по сети [2].

2. Обзор исследований в области обфускации программного кода. Обфускацией программы называется всякое ее преобразование, которое сохраняет вычисляемую программой функцию, но при этом придает программе такую форму, что извлечение из текста программного кода ключевой информации об алгоритмах и структурах данных, реализованных в этой программе, становится трудоемкой задачей [3].

Обфусцированной программой называется программа, которая после применения обфусцирующих преобразований, на всех допустимых для исходной программы входных данных выдает тот же самый результат, что и оригинальная программа, но более трудна для анализа, понимания и модификации [2].

В настоящее время исследования в области обфускации программного кода проводятся по двум направлениям [3]:

– системное программирование;

– математическая криптография.

С позиции системного программирования обфускация программы может использоваться для защиты авторских прав на программное обеспечение, для предотвращения реверс инжиниринга программ, для создания и защиты водяных знаков, обеспечения безопасности мобильных агентов в информационных сетях, для проведения безопасного поиска в потоках данных и защиты баз данных. Однако существенным недостатком данного подхода является отсутствие обоснования гарантированной стойкости. В случае применения методов динамического анализа программ и привлечения квалифицированных экспертов

в области системного программирования стойкости существующих средств обфускации программ оказывается недостаточно.

С позиции математической криптографии разработка эффективных алгоритмов позволит решить целый ряд серьезных вопросов, например, с ее помощью можно преобразовать криптосистему с секретным ключом к криптосистеме с открытым ключом, проводить вычисления над зашифрованными данными, реализовывать системы функционального шифрования, доверенные схемы перешифрования и электронно-цифровой подписи, создавать верифицируемые системы тайного голосования и схемы двойственного шифрования.

Для построения эффективного метода защиты файлов документальных форматов, внедренных в защищенный контейнер, предлагается использовать математический аппарат неразличимой обфускации программного кода, активно развивающийся в настоящее время в рамках направления математической криптографии [4]. Исследования в области неразличимой обфускации, проводимые в настоящее время, как российскими учеными (Варнавский Н.П., Захаров В.А., Кузюрин Н.Н.), так и зарубежными (S. Garg, C. Gentry, S. Halevi, B. Barak, J.S. Coron, T. Lepoint, M. Tibouchi) базируются на возможности обфускации булевых функций. Процедуру проверки прав доступа пользователя к документу, внедренному в контейнер, можно рассматривать как точечную функцию, поскольку результатом ее выполнения является значение из множества {0,1}, поэтому применение неразличимой обфускации для защиты данной проверки является также корректным. Данная работа посвящена модификации существующих подходов к осуществлению неразличимой обфускации с целью устранения ряда ограничений, обусловленных применяемыми механизмами, моделями и алгоритмами [5, 6].

3. Описание аналитической модели защиты файлов документальных форматов от несанкционированного доступа. Для описания предлагаемой математической модели защиты от несанкционированного доступа к файлам документальных форматов, отличающейся применением процедуры неразличимой обфускации программного кода, необходимо ввести ряд обозначений:

iO – обфускатор неразличимости (неразличимый обфускатор);

$f(x)$ – булева функция, принимающая на вход вектор x длины n;

$C(x)$ – булева схема, принимающая на вход вектор x длины n;

$\mathbb{N}$ – множество натуральных чисел;

BP_f – ветвящаяся программа;

MBP_f – матричная ветвящаяся программа;

$M\widetilde{B}P_f$ – рандомизированная матричная ветвящаяся программа;

p – большое простое число длиной $\Omega(n) > 512$ бит;

Z_p – кольцо вычетов по модулю p;

L – длина ветвящейся программы BP_f;

W – ширина ветвящейся программы BP_f.

Формально, обфускатор неразличимости iO можно считать компилятором, принимающим на вход булеву функцию $f(x)$ и генерирующим на выходе обфусцированную программу $f'(x) = iO(f(x))$. При этом должно выполняться следующее условие: $\forall x : f'(x) = f(x)$.

Введем понятие неразличимой обфускации для булевых схем класса NC^1, который представляет собой класс булевых функций, вычисляемых схемами из функциональных элементов полиномиальной сложности и логарифмической глубины $O(\log^1 n)$, где n – длина входного вектора.

Определение 1. Вычислительная неразличимость [7].

Пусть $\{X_n\}_n$ и $\{Y_n\}_n$ множества распределений вероятностей над множеством $\{0,1\}^{p(n)}$ для некоторого полинома $p(\cdot)$. Тогда $\{X_n\}_n$ и $\{Y_n\}_n$ вычислительно неразличимы ($\{X_n\}_n \approx \{Y_n\}_n$), если для любой неоднородной полиномиальной вероятностной машины Тьюринга D, существует пренебрежимо малая функция α такая, что $\forall n \in \mathbb{N}$:

$$\Pr[t \leftarrow X_n, D(t) = 1] - \Pr[t \leftarrow Y_n, D(t) = 1] \leq \alpha(n). \quad (1)$$

Определение 2. Обфускатором неразличимости iO называется однородная полиномиальная вероятностная машина Тьюринга для булевых схем класса $\{C_\lambda\}$, где λ – параметр безопасности, если обеспечивается выполнение следующих свойств [4]:

– функциональная эквивалентность. Существует пренебрежимо малая функция $\mu(\lambda)$ такая, что для всех $\lambda \in \mathbb{N}$ и схемы $C \in C_\lambda$:

$$\forall x : \Pr[iO(C(x)) = C(x)] \geq 1 - \mu(\lambda); \quad (2)$$

– свойство виртуального черного ящика (стойкость). Для любой неоднородной полиномиальной вероятностной машины Тьюринга D (распознавателя), существует пренебрежимо малая функция α такая, что для всех $\lambda \in \mathbb{N}$ и любой пары эквивалентных схем $C_1, C_2 \in C_\lambda$, имеющих одинаковый размер, распределения вероятностей случайных величин $iO(C_1)$ и $iO(C_2)$ вычислительно неразличимы, то есть выполняется соотношение:

$$\forall x : \left| \Pr\left[D(iO(\lambda, C_1(x))) = 1\right] - \Pr\left[D(iO(\lambda, C_2(x))) = 1\right] \right| \le \alpha(\lambda) . \qquad (3)$$

Это означает, что не существует алгоритма распознавания обфускации более эффективного, чем обычное предположение, сделанное на основе анализа входов и выходов обфусцированной программы (функции или обфусцированного набора инструкций).

Неразличимость обфусцированных программ означает, что если две эквивалентные программы (одинакового размера) P_1, P_2 ,такие, что $\forall x : P_1(x) = P_2(x)$, а $iO-$ это обфускатор неразличимости, который принимает на вход программу $P(x)$ и на выходе генерирует новую программу $iO(P(x))$, то $iO(P_1(x))$ и $iO(P_2(x))$ будут вычислительно неразличимы:

$$\exists P_1(x), P_2(x). \forall x : P_1(x) = P_2(x) \to iO(P_1(x)) \approx iO(P_2(x)) . \qquad (4)$$

Согласно работам [4, 8, 9] процедура неразличимой обфускации включает в себя пять основных этапов.

1. Преобразование булевой функций в вид ветвящейся программы.
2. Преобразование ветвящейся программы в вид матричной ветвящейся программы.
3. Рандомизация матричной ветвящейся программы.
4. Кодирование рандомизированной матричной ветвящейся программы с помощью схемы дифференциального кодирования (Graded Encoding Scheme).
5. Вычисление обфусцированной функции.

Для более подробного рассмотрения аналитической модели неразличимой обфускации на рисунке 1 приведены основные этапы реализации неразличимой обфускации.

$$f \xrightarrow{1} BP_f \xrightarrow{2} MBP_f \xrightarrow{3} M\tilde{B}P_f \xrightarrow{4} iO(f) \xrightarrow{5} f(x)$$

Рис. 1. Основные этапы реализации неразличимой обфускации

Этап 1. Преобразование булевой функции в вид ветвящейся программы. На первом этапе производится преобразование булевой функции $f(x)$ в вид ветвящейся программы BP_f по теореме Сауэр-хоффа [9]. Ветвящиеся программы – это логические схемы, которые хорошо моделируют вычисления с помощью одного процессора, читающего не более одного бита информации в единицу времени.

Определение 3. Ветвящаяся программа для вычисления булевой функции $f(x)$ определяется как ориентированный ациклический граф, вершины которого разделены на слои $0, 1, ..., L$ (L – длина вет-

вящейся программы) таким образом, что выполняются следующие условия:

– из вершин слоя i ребра могут вести только в вершины слоя $i+1$;

– в слое номер 0 имеется единственная вершина, ее входная степень равна нулю;

– каждая вершина слоя L помечена 0 или 1, выходные степени вершин слоя L равны нулю;

– каждая вершина слоев $0,1,...,L-1$ помечена одной из переменных $\{x_1,...,x_n\}$, и имеет два выходящих ребра с пометками 0 и 1.

Процедура вычисления функции $f(x_1,...,x_n)$ с помощью такой программы состоит в том, что осуществляется проход от единственной вершины слоя 0 до некоторой вершины слоя L по ориентированным ребрам графа. Проходя через вершину, помеченную переменной x_i, считывается значение элемента входного вектора $x_i=0$ (или $x_i=1$) и далее осуществляется движение по выходящему ребру с соответствующей пометкой. Добравшись до вершины уровня L, рассчитывается значение функции $f(x_1,...,x_n)$ в соответствии с пометкой на финальной вершине.

Шириной W ветвящейся программы называется максимальное число вершин, находящихся на одном уровне. Размером V ветвящейся программы называется общее количество вершин графа на всех уровнях. На рисунке 2 представлен пример построения ветвящейся программы для вычисления функции $f(x_1,x_2,x_3)=(\overline{x}_1\wedge x_2\wedge\overline{x}_3)\vee(x_1\wedge\overline{x}_2\wedge x_3)$.

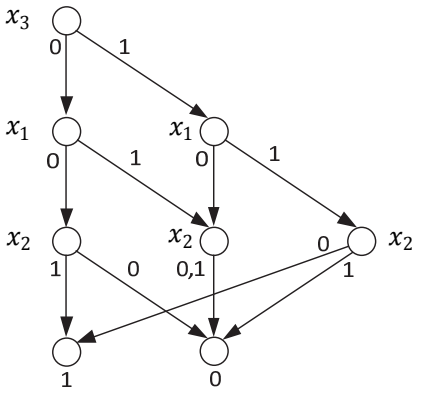

Рис. 2. Ветвящаяся программа для вычисления функции $f(x_1,x_2,x_3)=(\overline{x}_1\wedge x_2\wedge\overline{x}_3)\vee(x_1\wedge\overline{x}_2\wedge x_3)$

В работах [4, 11, 12] для преобразования булевой функции в вид ветвящейся программы применялась теорема Баррингтона. Теорема Баррингтона [13] гласит, что если для булевой функции $f(x)$ существует булева схема глубины L_C, то $f(x)$ можно вычислить ветвящейся программой шириной $W = 5$ и длиной $L \le 4^{L_C}$.

В предлагаемой аналитической модели для преобразования булевой функций в ветвящуюся программу предложено применять теорему Сауэрхоффа. Данный подход позволяет преобразовать любую схему размера V в ветвящуюся программу шириной $W \le 2(V+1)$ и длиной $L \le V$. Теорема Сауэрхоффа более эффективна при преобразовании булевых функций по сравнению с подходом на основе теоремы Баррингтона. Размер ветвящейся программы, полученной по теореме Баррингтона удовлетворяет следующему требованию $V \le L(f)^2$, где $L(f)$ – размер формулы булевой функции, а ветвящаяся программа, построенная по теореме Сауэрхоффа удовлетворяет следующему требованию $V \le 1.360L(f)^{\beta}$, где $\beta = \log_4(3+\sqrt{5}) < 1.195$ [10]. Применение теоремы Сауэрхоффа позволяет значительно сократить объем получаемых обфусцированных данных на выходе аналитической модели.

Этап 2. Преобразование ветвящейся программы в вид матричной ветвящейся программы. На данном этапе ветвящаяся программа BP_f преобразуется в функционально эквивалентную матричную ветвящуюся программу MBP_f.

Определение 4. Пусть $P_{rej} \in \{0,1\}^{W \times W}$ – класс матриц перестановок размерности $W \times W$, таких что $P_{rej} \ne I_{W \times W}$, где $I_{W \times W}$ – единичная матрица размерности $W \times W$. Матричная ветвящаяся программа шириной W и длиной L для входного вектора размером n бит определяется следующей последовательностью:

$$MBP_f = (I_{W \times W}, P_{rej}, inp(i), B_{i,0}, B_{i,1})_{i=1}^{L}, \quad (5)$$

где $B_{i,b} \in \{0,1\}^{W \times W}$ – матрицы перестановок, $b \in \{0,1\}$; $inp(i) : [L] \to [n]$ – функция выбора позиции матриц для текущего входного бита.

Преобразование ветвящейся программы в вид матричной ветвящейся программы осуществляется следующим образом: для каждого слоя $i \in [L]$ ветвящейся программы BP_f составляются перестановки и соответствующие матрицы перестановок $B_{i,0}, B_{i,1}$. При этом выбор матрицы $B_{i,b}$ определяется пометкой b выходящего ребра.

Результат выполнения булевой функции, представленной в виде матричной ветвящейся программы MBP_f, получаемый при входном векторе $x \in \{0,1\}^n$, определяется в соответствии с выражением (6).

$$MBP_f(x) = \begin{cases} 1, \text{ если } \prod_{i=1}^{L} B_{i,inp(i)} = I_{W \times W}; \\ 0, \text{ если } \prod_{i=1}^{L} B_{i,inp(i)} = P_{rej}. \end{cases} \quad (6)$$

Процедура преобразования ветвящейся программы в матричную ветвящуюся программу будет рассмотрена далее на примере перестановочной ветвящейся программы PBP_f фиксированной ширины [14]. На рисунке 3 представлена перестановочная ветвящаяся программа, для которой при входном векторе $x = (0,1,1)$ на выходе вычисляется соответствующая перестановка $P(x) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$.

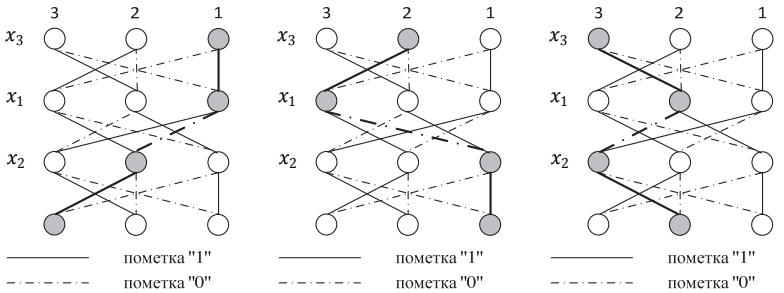

Рис. 3. Перестановочная ветвящаяся программа PBP_f шириной $W = 3$

Для построения матричной ветвящейся программы необходимо построить для первого слоя две перестановки $P_{1,1}, P_{1,0}$:

$$P_{1,1}(x_3) = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}; \qquad P_{1,0}(x_3) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

При этом матрицы перестановок $B_{1,1}, B_{1,0}$, соответствующие перестановкам $P_{1,1}, P_{1,0}$ будут иметь следующий вид:

$$B_{1,1=}\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix};\qquad B_{1,0=}\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}.$$

Затем рассчитываются перестановки и соответствующие им матрицы перестановок для второго и третьего слоя:

$$P_{2,1}(x_1)=\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix};\qquad P_{2,0}(x_1)=\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix};$$

$$B_{2,1=}\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix};\qquad B_{2,0=}\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix};$$

$$P_{3,1}(x_2)=\begin{pmatrix}1&2&3\\1&3&2\end{pmatrix};\qquad P_{3,0}(x_2)=\begin{pmatrix}1&2&3\\3&2&1\end{pmatrix};$$

$$B_{3,1=}\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix};\qquad B_{3,0=}\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}.$$

Для проверки корректности построенных матриц необходимо рассчитать $\prod_{i=1}^{L} B_{i,inp(i)}$ для входного вектора $x=(0,1,1)$:

$$\prod_{i=1}^{3} B(0,1,1)=B_{1,1}B_{2,0}B_{3,1}=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}=\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}. \quad (7)$$

Из выражения (6) следует, что полученная в выражении (7) матрица соответствует исходной перестановке $P(x)=\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}$. Полученная матрица $\prod_{i=1}^{3} B(0,1,1)=P_{rej}$, соответственно $MBP_f(0,1,1)=0$.

При значении входного вектора $x=(0,1,0)$: $MBP_f(0,1,0)=1$ поскольку:

$$\prod_{i=1}^{3} B(0,1,0) = B_{1,0}B_{2,0}B_{3,1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Этап 3. Рандомизация матричной ветвящейся программы. Для сокрытия базового алгоритма вычисления и значений матриц применяется процедура рандомизации по методу Килиана [15]. Интерпретация протокола скрытой передачи Килиана в аспекте неразличимой обфускации позволила повысить стойкость неразличимой обфускации, в особенности при защите от таких атак как: "атаки с частичным раскрытием", "атаки смешенных входных данных" [4].

Определение 5. Процедура рандомизации матричной ветвящейся программы $rand_p(MBP_f)$ над кольцом Z_p осуществляется следующим образом:

– выбор случайных независимых скалярных значений $\{\alpha_{i,0}, \alpha'_{i,1}, \alpha'_{i,0}, \alpha'_{i,1} \in Z_p : i \in [L]\}$ над кольцом Z_p, таким образом, что $\prod_{i\in I_j} \alpha_{i,0} = \prod_{i\in I_j} \alpha'_{i,0}$ и $\prod_{i\in I_j} \alpha_{i,1} = \prod_{i\in I_j} \alpha'_{i,1}$, где $I_j := \{i \in [L] : inp(i) = j\}$, $j \in [n]$ – набор слоев ветвящейся программы и соответствующий входной бит i;

– для каждого из слоев ветвящейся программы $i \in [L]$ рассчитываются четыре матрицы $D_{i,0}, D_{i,0}, D'_{i,0}, D'_{i,1}$ размерности $(2L+W)\times(2L+W)$ так, что:

$$D_{i,b} = \begin{bmatrix} d_{i,b} & 0 \\ 0 & \alpha_{i,b}B_{i,b} \end{bmatrix}, \qquad D'_{i,b} = \begin{bmatrix} d'_{i,b} & 0 \\ 0 & \alpha'_{i,b}I_{W\times W} \end{bmatrix},$$

где $d_{i,b}, d'_{i,b}$ – случайная диагональная матрица по модулю p размерности $2L\times 2L$, $b \in \{0,1\}$;

– выбор векторов s,t и s',t' размерности $(2L+W)$ таких, что выполняются следующие условия:

$s = (\bar{0}, \bar{s}_R, \hat{s})$, где $\bar{0}$ – нулевой вектор длины L, $\bar{s}_R$ – случайный вектор длины L, $\hat{s}$ – случайный вектор длины W;

$t = (\bar{t}_R, \bar{0}, \hat{t})^T$, где $\bar{0}$ – нулевой вектор длины L, $\bar{t}_R$ – случайный вектор длины L, $\hat{t}$ – случайный вектор длины W;

$s' = (\overline{0}, \overline{s'_R}, \hat{s'})$, где $\overline{0}$ – нулевой вектор длины L, $\overline{s'_R}$ – случайный вектор длины L, $\hat{s'}$ – случайный вектор длины W;

$t' = (\overline{t'_R}, \overline{0}, \hat{t'})^T$, где $\overline{0}$ – нулевой вектор длины L, $\overline{t}_R$ – случайный вектор длины L, $\hat{t}$ – случайный вектор длины W;

$\langle \hat{s}, \hat{t} \rangle = \langle \hat{s'}, \hat{t'} \rangle$ – скалярные произведения элементов векторов s, t и s', t' равны.

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

– вычисление матриц $\widetilde{D}_{i,b} := R_{i-1} D_{i,b} (R_i)^{-1}$ и $\widetilde{D}'_{i,b} := R'_{i-1} D'_{i,b} (R'_i)^{-1} \quad \forall i \in [n], b \in \{0,1\}$;

– вычисление рандомизированных значений векторов $\widetilde{s} = s \cdot R_0^{-1}$; $\widetilde{t} = R_L \cdot t$ и $\widetilde{s'} = s' \cdot (R'_0)^{-1}$; $\widetilde{t'} = R'_L \cdot t'$.

Таким образом, результатом данного этапа является рандомизированная матричная ветвящаяся программа над кольцом Z_p, которая представляет собой следующую совокупность:

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

Этап 4. Кодирование рандомизированной матричной ветвящейся программы с помощью схемы дифференциального кодирования. Для кодирования рандомизированных матриц ветвящейся программы можно применяться схема дифференциального кодирования, которая описана в [16, 17]. Предложенная схема обладает свойством гомоморфного шифра [18]: обеспечивает конфиденциальность значений каждой из матриц при возможности осуществления математических операций над закодированными данными без их раскодирования. Разрешенными операциями над зашифрованными данными при этом являются произведение и сложение, которые удовлетворяют следующему соотношению:

$$\begin{cases} E(m_1) \otimes E(m_2) = E(m_1 \cdot .m_2), \\ E(m_1) \oplus E(m_2) = E(m_1 + m_2); \end{cases}$$

где $\otimes$ и $\oplus$ – операции над шифрованными данными, соответствующие операциям умножения и сложения над открытыми данными; $E(m_1), E(m_2)$ – шифр тексты сообщений m_1 и m_2 соответственно.

Однако в отличие от полностью гомоморфного шифрования в схеме дифференциального кодирования существуют ограничения на операции сложения и умножения. Каждая зашифрованная матрица кодируется относительно некоторого множества, являющегося подмножеством универсума $S \in [U]$. При этом две матрицы может складывать только, если они кодируются по отношению к одному множеству S. В свою очередь умножение возможно только, если шифрование осуществляется относительно двух непересекающихся множеств S и S'.

Определение 6. Система разделенных множеств представляет собой совокупность множеств $S_n = \{S_{i,b} : i \in [n], b \in \{0,1\}\}$ из $2n$ элементов над универсумом $U = \{1,2,...,U_{\max}\}$, таких что $\bigcup_{i\in[n]} S_{i,0} = \bigcup_{i\in[n]} S_{i,1} = U$, $S_{i,0} = \bigcup_{k=1}^{|ind(i)|} S_{i,0}^k$, $S_{i,1} = \bigcup_{k=1}^{|ind(i)|} S_{i,1}^k$ при этом $\forall i_1, i_2, k_1, k_2 : S_{i_1,b}^{k_1} \cap S_{i_2,b}^{k_2} = \varnothing$, $b \in \{0,1\}$, $|ind(i)|$ – частота встречаемости i-го входного бита в $M\widetilde{B}P_f$ программе длиной L. Значение $U_{\max}$ рассчитывается по следующей формуле: $U_{\max} = \sum_{i=1}^{n} (2 \cdot |ind(i)| - 1)$.

Предложенная система разделенных множеств позволяет осуществить полное покрытие универсума U в случае корректного умножения матриц – структура ветвящейся программы хранит взаимосвязи входных бит и слоев программы (соответствующих матриц). В выражении (8) представлен предлагаемый подход к построению системы разделенных множеств в рамках одного i-го входного бита при $k = |ind(i)|$:

$$\begin{aligned} &S_{1,0} = \{1\}, S_{2,0} = \{2,3\}, ..., S_{k-1,0} = \{2k-4, 2k-3\}, S_{k,0} = \{2k-2, 2k-1\}; \\ &S_{1,1} = \{1,2\}, S_{2,1} = \{3,4\}, ..., S_{k-1,1} = \{2k-3, 2k-2\}, S_{k,1} = \{2k-1\}. \end{aligned} \quad (8)$$

Пример построения системы разделенных множеств представлен на рисунке 4.

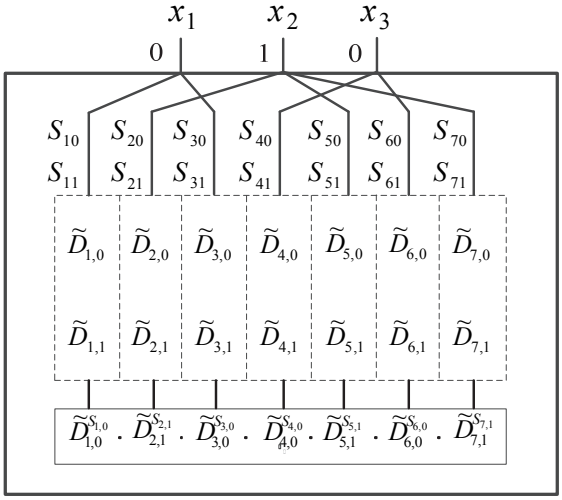

Рис. 4. Пример построения системы разделенных множеств

Пусть на вход ветвящейся программы длиной $L = 7$ поступает вектор $x = (0,1,0)$. При этом $|ind(x_1)| = 2$, $|ind(x_2)| = 3$, $|ind(x_3)| = 2$. Построение системы осуществляется следующим образом: последовательно для каждого входного бита задаются элементы подмножеств S таким образом, чтобы выполнялось условие $\bigcup_{i\in[n]} S_{i,0} = \bigcup_{i\in[n]} S_{i,1} = U$:

$$x_1 :\ S_{1,0} = \{1\}, S_{3,0} = \{2,3\}, S_{1,1} = \{1,2\}, S_{3,1} = \{3\};$$

$$x_2 :\ S_{2,0} = \{4\}, S_{5,0} = \{5,6\}, S_{7,0} = \{7,8\}, S_{2,1} = \{4,5\}, S_{5,1} = \{6,7\}, S_{7,1} = \{8\};$$

$$x_3 :\ S_{4,0} = \{9\}, S_{6,0} = \{10,11\}, S_{4,1} = \{9,10\}, S_{6,1} = \{11\}.$$

Предложенная система разделенных множеств позволяет защититься от атак с использованием смешанных произведений. Злоумышленнику не известны взаимосвязи входных бит и соответствующих матриц, за счет осуществления смешанных произведений он может попытаться восстановить эти взаимосвязи. Но без их точного знания злоумышленник не сможет осуществить полное покрытие универсума U, а тем самым и вычисление зашифрованной функции.

Определение 7. Для $k+1$ мультипликативных циклических групп $G_1,...,G_k,G_T$, конечного порядка p, существует k–линейное отображение $e: G_1 \times ... \times G_k \to G_T$, если выполняются следующие свойства:

– для всех элементов $g_1 \in G_1, ..., g_k \in G_k, i \in [k]$ и целого $\alpha \in Z_p$:

$$e(g_1, ..., \alpha \cdot g_i, ..., g_k) = \alpha \cdot e(g_1, ..., g_k);$$

– если элементы $g_i \in G_i, i \in [k]$, являются образующими элементами групп G_i, то $e(g_1, ..., \alpha \cdot g_i, ..., g_k)$ является образующим элементом группы G_T.

В настоящие время известны два наиболее общих подхода к построению k–линейных отображений, реализующих схему дифференциального кодирования. Первый из них [15] базируется на полилинейных отображениях на решетках, а второй [16] – для целых чисел. Предлагаемая аналитическая модель базируется на втором подходе, поскольку реализация полилинейных отображений на основе целых чисел является более эффективной в сравнении с применением решеток [16].

Определение 8. Схемой дифференциального кодирования называется совокупность вероятностных полиномиальных алгоритмов $(InstGen, Enc, Add, Mult, isZero)$, которые выполняют следующие функции:

– Instance Generation: $(sp, pp) \leftarrow InstGen(1^\lambda, 1^k)$.

$InstGen$ принимает на вход λ (параметр безопасности) и k (порядок полилинейности отображения). На выходе генерируется секретный параметр sp и открытый параметр pp. Секретный параметр sp содержит целое число Y, такое что $k \le Y \le 2k$, простые числа $g_1, ..., g_N$, где $N = \eta \log_2 \lambda$, $\eta = (k+1)(5\lambda+2)+6$ [16] и совокупность множеств $\{E_S^m : m \in Z_{g_1} \times ... \times Z_{g_N}, S \subseteq [U]\}$. E_S^m рассматривается как набор возможных кодирований значений m относительно множества S.

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

Enc принимает на входе секретный параметр sp, значение открытого сообщения $m \in Z_{g_1} \times ... \times Z_{g_N}$, и множество $S \subseteq [U]$, на выходе

осуществляется шифрование m относительно множества S (обозначение – $u \in E_S^m$).

– Addition: $u \leftarrow Add(pp, u, u')$.

Add принимает на входе открытый параметр pp и зашифрованные сообщения $u \in E_S^m$, $u' \in E_{S'}^{m'}$, на выходе вычисляется значение зашифрованного сообщения равного сумме двух исходных зашифрованных сообщений $u \in E_S^{m+m'}$, в том случае, если $S = S'$.

– Multiplication: $u \leftarrow Mult(pp, u, u')$.

$Mult$ принимает на входе открытый параметр pp и зашифрованные сообщения $u \in E_S^m$, $u' \in E_{S'}^{m'}$, на выходе вычисляется значение зашифрованного сообщения равного произведению двух исходных зашифрованных сообщений $u \in E_{S \bigcup S'}^{m \cdot m'}$, в том случае, если $S \bigcap S' = \varnothing$.

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

$isZero$ принимает на входе открытый параметр pp и зашифрованное сообщение u, на выходе вычисляется значение, характеризующее открытое сообщение, 1 – в случае, если $u \in E_{[U]}^0$ (кодирование нуля), и 0 – в остальных случаях.

Далее представлено описание схемы дифференциального кодирования для шифрования рандомизированной матричной ветвящейся программы в рамках аналитической модели. Для программы длиной L необходима реализация $(L+2)$ – линейного отображения, поскольку L – число умножений элементов входного вектора в программе, но также необходимо учесть умножение на вектора s, t. При этом основные операции схемы дифференциального кодирования принимают следующий вид:

– $InstGen$: пусть $Y = L + 2$ – требуемый размер системы разделенных множеств. Изначально выбираются N секретных больших простых чисел $\{p_i\}_{i=1}^N$, затем вычисляется их произведение $x_0 = \prod_{i=1}^N p_i$. Также выбираются N секретных простых чисел $\{g_i\}_{i=1}^N$, N случайных целых чисел $\{h_i\}_{i=1}^N$, N случайных целых чисел $\{r_i\}_{i=1}^N$ и Y случайных значений $z_1, ..., z_Y \in Z_{x_0}$. Затем осуществляется вы-

числение параметра процедуры Zero Test p_{zt}, заданного целым числом, в соответствии с выражением (9):

$$p_{zt} = \sum_{i=1}^{N} h_i.(\prod_{j=1}^{z} z_j.g_i^{-1} \bmod p_i).\prod_{i' \neq i} p_{i'} \bmod x_0 \ . \qquad (9)$$

Выходными для данной процедуры являются следующие параметры: секретные – $(\{z_i\}_{i=1}^{Y}, \{g_i\}_{i=1}^{N}, \{p_i\}_{i=1}^{N})$, открытые – (p_{zt}, x_0).

– *Enc*: пусть $u \in Z_{x_0}$ является кодированием сообщения $m = (m_1,...,m_N) \in Z_{g_1} \times ... \times Z_{g_N}$ относительно множества $S \subseteq [U]$, тогда для всех $1 \le i \le N$ и случайных (малых) целых чисел r_i:

$$u \equiv \frac{r_i.g_i + m_i}{\prod_{j \in S} z_j} (\bmod p_i) \ .$$

– *Add*: при заданных $u, u' \in Z_{x_0}$, где

$$\forall i : u \equiv \frac{r_i.g_i + m_i}{\prod_{j \in S} z_j} (\bmod p_i), \qquad u' \equiv \frac{r_i'.g_i + m_i'}{\prod_{j \in S} z_j} (\bmod p_i) \ ,$$

справедливо следующее выражение:

$$\forall i : u + u' \equiv \frac{(r_i + r_i').g_i + (m_i + m_i')}{\prod_{j \in S} z_j} (\bmod p_i) \ .$$

– *Mult*: Пусть S и S' являются множествами такими, что $S \bigcap S' = \varnothing$. Тогда при заданных $u, u' \in Z_{x_0}$, где

$$\forall i : u \equiv \frac{r_i.g_i + m_i}{\prod_{j \in S} z_j} (\bmod p_i), \qquad u' \equiv \frac{r_i'.g_i + m_i'}{\prod_{j \in S'} z_j} (\bmod p_i) \ ,$$

справедливо следующее выражение:

$$\forall i : u \cdot u' \equiv \frac{(r_i r_i' + r_i m_i' + r_i' m_i) g_i + m_i m_i'}{\prod_{j \in S \bigcup S'} z_j} (\bmod p_i) \ ,$$

где $\forall i : (r_i r_i' + r_i m_i' + r_i' m_i) g_i + m_i m_i' \le p_i$.

Согласно представленной схеме для дифференциального кодирования рандомизированной матрицы ветвящейся программы $M\widetilde{B}P_f$ необходима реализация $(L+2)$ – уровневой схемы кодирования. При этом также необходимо сгенерировать:

– r_s, r_t, r'_s, r'_t – случайные вектора, которые используются для кодирования векторов $\widetilde{s}, \widetilde{t}, \hat{s}', \hat{t}'$;

– $U_{i,b}, U'_{i,b}$ – случайные матрицы ($i \in [L], b \in \{0,1\}$).

Пусть $U_s, U_t, U_1, ..., U_L$ – непересекающиеся множества универсума U такие, что $U = U_s \bigcup U_t \bigcup \bigcup_{j=1}^{L} U_j$. Аналогично и для множеств, образующих универсум $U = U'_s \cup U'_t \cup \bigcup_{j=1}^{n} U'_j$. При этом U_s, U_t и U'_s, U'_t используются для кодирования $\widetilde{s}, \widetilde{t}$ и $\widetilde{s}', \widetilde{t}'$ соответственно.

Таким образом, результатом данного этапа является обфусцированная функция $iO(f(x))$, которая представляет собой следующую совокупность:

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

где $g = CRT(g_i), g_i \le p_i$ – вычисляется по китайской теореме об остатках [19].

Этап 5. Вычисление обфусцированной булевой функции.

Вычисление значения обфусцированной булевой функций $iO(f(x))$ при заданном входном векторе x осуществляется с помощью процедур *Add* и *Mult*, которые в свою очередь позволяют вычислить значение выражения (10). Данное выражение представляет собой кодирование разности произведений рандомизированных матриц $\widetilde{D}$ и $\widetilde{D}'$:

$$Enc(\widetilde{s}.\prod_{i=1}^{L} \widetilde{D}_{i,inp(i)}.\widetilde{t} - \widetilde{s}'.\prod_{i=1}^{L} \widetilde{D}'_{i,inp(i)}.\widetilde{t}') = u\,. \qquad (10)$$

Пусть

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

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

Если B является единичной матрицей, то согласно выражению (6) и условию $\langle \hat{s}, \hat{t} \rangle = \langle \hat{s}', \hat{t}' \rangle$:

$$q - q' = \hat{s} \cdot B \cdot \hat{t}^T - \hat{s}' \cdot I_{W \times W} \cdot \hat{t}'^T = \hat{s} \cdot I_{W \times W} \cdot \hat{t}^T - \hat{s}' \cdot I_{W \times W} \cdot \hat{t}'^T = 0. \qquad (11)$$

Из выражения (11) следует, что $u = Enc(q - q') = Enc(0)$ является кодированием 0.

Для непосредственного вычисления значения обфусцированной булевой функции применяется операция $isZero$. Если $isZero(pp, u) = 1$, то $f(x) = 1$, иначе $f(x) = 0$. На рисунке 5 представлена иллюстрация процедуры вычисления обфусцированной функции при определенном входном векторе x.

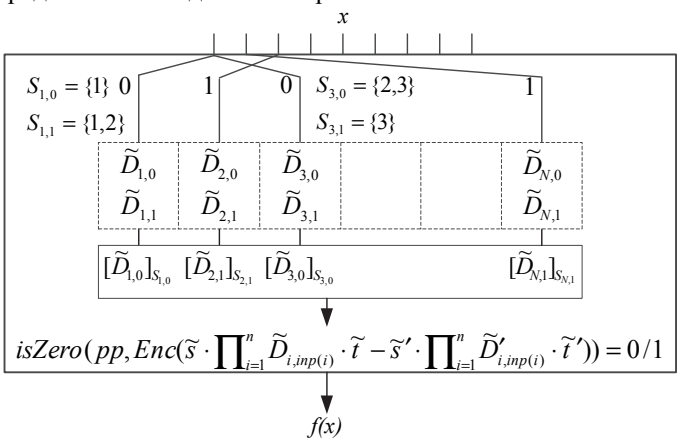

Рис 5. Процедура вычисления обфусцированной булевой функции при определенном входном векторе x

Если u является кодированием нуля $\forall i \in [N]: m_i = 0$, то, исходя из предыдущего выражения, следует:

$$p_{zt} \cdot u = \sum_{i=1}^{N} h_i r_i \cdot \prod_{i' \neq i} p_{i'} (\mathrm{mod}\, x_0). \qquad (13)$$

Поскольку $h_i.r_i << p_i$, то $h_i.r_i.p_{i'} = h_i.r_i \cdot \frac{x_o}{p_i} << x_0$, а также сумма $\sum_{i=1}^{N} h_i r_i p_{i'} << x_0$. Из выражения (12) следует, что $\omega = p_{zt} \cdot u (\mathrm{mod}\ x_0)$ имеет малое значение по сравнению с x_0, тогда произведения h_i, r_i должны удовлетворять определенным ограничениям по размеру. Поэтому согласно лемме 3 из [16], при $m = 0$ – $|\omega| < x_0 \cdot 2^{-\upsilon-\lambda-2}$, а если $m \neq 0$, то $|\omega| \geq x_0 \cdot 2^{-\nu+2}$, где $\upsilon = \alpha + \beta + 11$, α – размерность в битах простых чисел $\{g_i\}_{i=1}^{N}$, β – размерность в битах случайных целых чисел $\{h_i\}_{i=1}^{N}$. Таким образом:

$$isZero(pp,u) = \begin{cases} 1, \text{ если } |\omega| < x_0.2^{-\upsilon-\lambda-2}; \\ 0, \text{ если } |\omega| \geq x_0.2^{-\upsilon+2}. \end{cases} \quad (14)$$

Из выражения (14) следует, что значение обфусцированной функции будет вычисляться следующим образом:

$$O(f(x)) = \begin{cases} 1, \text{если } isZero = 1; \\ 0, \text{если } isZero = 0. \end{cases} \quad (15)$$

4. Оценка свойств аналитической модели защиты файлов документальных форматов от несанкционированного доступа. Разработанная модель предназначена для описания этапов, необходимых для осуществления неразличимой обфускации программного кода, с целью защиты файлов документальных форматов от несанкционированного доступа.

Согласно общепринятым требованиям модель должна удовлетворять следующим свойствам: адекватность (соответствие модели исходной реальной системе и учет, наиболее важных качеств, связей и характеристик), точность (степень совпадения полученных в процессе моделирования результатов с характеристиками реального объекта), универсальность (применимость модели к анализу ряда однотипных систем в одном или нескольких режимах функционирования), целесообразность (точность получаемых результатов и общность решения задачи должны увязываться с затратами на моделирование) [20].

Адекватность. Разработанная модель позволяет адекватно представить все элементы, необходимые для осуществления неразли-

чимой обфускации программного кода, определить требования к ним так, чтобы они удовлетворяли требованиям по стойкости и функциональной эквивалентности обфусцированной программы.

Точность. Точность представленной модели характеризуется выполнением предъявленных к ней требований по стойкости и функциональной эквивалентности обфусцированной программы.

Универсальность. Представленная модель подходит для всех видов файлов документальных форматов и любых возможных видов угроз.

Целесообразность. Для осуществления защиты файлов документальных форматов от несанкционированного доступа на основе математического аппарата неразличимой обфускации представленная модель является необходимой.

Экспериментальная проверка модели и формальное доказательство стойкости разработанного подхода является направлением дальнейших исследований.

5. Заключение. Разработанная аналитическая модель защиты от несанкционированного доступа к файлам документальных форматов, отличающаяся применением процедуры неразличимой обфускации программного кода позволяет реализовать метод контролируемого разграничения доступа в автоматизированной системе, обрабатывающей информацию различного уровня конфиденциальности, при условии разрешения одновременного доступа пользователя к ресурсам различного уровня конфиденциальности.

Литература

1. *Козачок А.В., Туан Л.М.* Обоснование возможности применения неразличимой обфускации для защиты исполняемых файлов // Перспективные информационные технологии: Сб. тр. междунар. НТК. Самара: 2015. Т. 1. С. 269–272.
2. *Козачок А.В., Туан Л.М.* Комплекс алгоритмов контролируемого разграничения доступа к данным, обеспечивающий защиту от несанкционированного доступа // Системы управления и информационные технологии. 2015. № 3(61). С. 58–61.
3. *Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.А.* Современное состояние исследований в области обфускации программ: определения стойкости обфускации // Труды Института Системного программирования: М.: ИСП РАН. 2014. Т. 26. № 3. С. 167–198.
4. *Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B.* Candidate indistinguishability obfuscation and functional encryption for all circuits // In FOCS. 2013. pp. 40–49.
5. *Варновский Н.П., Захаров В.А., Кузюрин Н.Н.* Математические проблемы обфускации // Математика и безопасность информационных технологий. Материалы конференции в МГУ. Москва: 2005. С. 65–91.
6. *Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vadhan S.P., Yang K.* On the (im)possibility of program obfuscation // In Advances in Cryptology – CRYPTO. 2001. pp. 1–18.

7. *Pass R., Shelat A.* A Course in Cryptograph // Theoretical Foundation of Cryptography. 2010. pp. 68–71.
8. *Ananth P., Gupta D., Ishai Y., Sahai A*. Optimizing obfuscation: Avoiding barrington's theorem // In Proceeding of the 2014 ACM SIGSAC. 2014. pp. 646–658.
9. *Pass R., Seth K., Telang S*. Indistinguishability obfuscation from semantically-secure multilinear encodings // In Advances in Cryptology – CRYPTO. 2014. pp. 500–517.
10. *Sauerhoff M., Wegener I., Werchner R.* Relating branching program size and formula size over the full binary basis // In Proceedings of 16th STACS. 1999. pp. 57–67.
11. *Barak B., Garg S., Kalai Y. T., Paneth O., Sahai A*. Protecting obfuscation against algebraic attacks // In Advances in Cryptology – EUROCRYPT. 2014. pp. 221–238.
12. *Brakerski Z., Rothblum G. N.* Virtual black-box obfuscation for all circuits via generic graded encoding // In Proceedings of 11th Theory of Cryptography Conference, TCC. 2014. pp. 1–25.
13. *Barrington D. A*. Bounded-width polynomial-size branching programs recognize exactly those languages in NC1 // Journal of Computer and System Sciences. 1989. pp. 150–164.
14. *Balaji N., Krebs A., Limaye N*. Skew Circuits of Small Width // In Proceedings of 21st International Conference, COCOON. Beijing. 2015. pp. 199–210.
15. *Kilian J*. Founding cryptography on oblivious transfer // In 20th Annual ACM Symposium on Theory of Computing. 1988. pp. 20–31.
16. *Garg S., Gentry C., Halevi S*. Candidate multilinear maps from ideal lattices // In Advances in Cryptology – EUROCRYPT. 2013. pp. 1–17.
17. *Coron J. S., Lepoint T., Tibouchi M*. Practical multilinear maps over the integers // In In Advances in Cryptology – CRYPTO 2013. pp. 476–493.
18. *Варновский Н.П., Шокуров А.В*. Гомоморфное шифрование // Труды Института Системного программирования: М.: ИСП РАН. 2006. Т. 12. С. 27–36.
19. *Нестеренко Ю.В*. Теория чисел // М.: Академия. 2008. 273 с.
20. *Бабенко Л.К., Буртыка Ф.Б., Макаревич О.Б., Трепачева А.В.* Обобщенная модель системы криптографически защищенных вычислений // Известия ЮФУ. Технические науки. 2015. № 5(166). С. 77–86.

References

1. Kozachok A.V., Tuan L.M. [On the possibility of application indistinguishability obfuscation to protect executable files] *Perspektivnye informacionnye tehnologii: Sb. tr. mezhdunar. NTK* [Advanced information technologies: Proceedings of the International Conference]. Samara: 2015. vol. 1. pp. 269–272. (In Russ.).
2. Kozachok A.V., Tuan L.M. [Controlling data access restriction algorithm providing protection from unauthorized access] *Sistemy upravlenija i informacionnye tehnologii – Control Systems and Information Technologies*. 2015. vol. 3(61). pp. 58–61. (In Russ.).
3. Varnovskij N.P., Zaharov V.A., Kuzjurin N.N., Shokurov A.V. [Modern state of research in program obfuscation: the definition of obfuscation security]. *Trudy Instituta Sistemnogo programmirovanija – Proceedings of the Institute for System Programming*. M.: ISP RAS. 2014. vol. 26(3). pp. 167–198. (In Russ.).
4. Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B. Candidate indistinguishability obfuscation and functional encryption for all circuits. *In FOCS*. 2013. pp. 40–49.
5. Varnovskij N.P., Zaharov V.A., Kuzjurin N.N. [Mathematical problems of obfuscation]. *Matematika i bezopasnost' informacionnyh tehnologij. Materialy konferencii v MGU* [Mathematics and security of information technology. Proceedings of the conference at Moscow State University]. Moscow: 2005. pp. 65–91. (In Russ.).

6. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vadhan S.P., Yang K. On the (im)possible obfuscating programs. In Advances in Cryptology – CRYPTO. 2001. pp. 1–18.
7. Pass R., Shelat A. A Course in Cryptograph. Theoretical Foundation of Cryptography. 2010. pp. 68–71.
8. Ananth P., Gupta D., Ishai Y., Sahai A. Optimizing obfuscation: Avoiding barrington's theorem. In Proceeding of the 2014 ACM SIGSAC. 2014. pp. 646–658.
9. Pass R., Seth K., Telang S. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Advances in Cryptology – CRYPTO. 2014. pp. 500–517.
10. Sauerhoff M., Wegener I., Werchner R. Relating branching program size and formula size over the full binary basis. In Proceedings of 16th STACS. 1999. pp. 57–67.
11. Barak B., Garg S., Kalai Y. T., Paneth O., Sahai A. Protecting obfuscation against algebraic attacks. In Advances in Cryptology – EUROCRYPT 2014. pp. 221–238.
12. Brakerski Z., Rothblum G. N. Virtual black-box obfuscation for all circuits via generic graded encoding. In Proceedings of 11th Theory of Cryptography Conference, TCC. 2014. pp. 1–25.
13. Barrington D. A. Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. *Journal of Computer and System Sciences*. 1989. pp. 150–164.
14. Balaji N., Krebs A., Limaye N. Skew Circuits of Small Width. In Proceedings of 21st International Conference, COCOON. Beijing. 2015. pp. 199–210.
15. Kilian J. Founding cryptography on oblivious transfer. In 20th Annual ACM Symposium on Theory of Computing. 1988. pp. 20–31.
16. Garg S., Gentry C., Halevi S. Candidate multilinear maps from ideal lattices. In Advances in Cryptology – EUROCRYPT. 2013. pp. 1–17.
17. Coron J. S., Lepoint T., Tibouchi M. Practical multilinear maps over the integers. In In Advances in Cryptology – CRYPTO. 2013. pp. 476–493.
18. Varnovskij N.P., Shokurov A.V. [Homomorphic encryption]. *Trudy Instituta Sistemnogo programmirovanija – Proceedings of the Institute for System Programming*. M.: ISP RAS. 2006. vol. 12. pp. 27–36. (In Russ.).
19. Nesterenko Ju.V. *Teorija chisel* [Number theory]. M. Academy. 2008. 273 p. (In Russ.).
20. Babenko L.K., Burtyka F.B., Makarevich O.B., Trepacheva A.V. [The generalized model of a cryptographically secure computing]. *Izvestiya YuFU. Tekhnicheskie nauki – Izvestiya SFedU. Engineering Sciences*. 2015. vol. 5(166), pp. 77-86. (In Russ.).

Козачок Александр Васильевич — к-т техн. наук, сотрудник, Академия Федеральной службы охраны Российской Федерации. Область научных интересов: информационная безопасность, защита от несанкционированного доступа, математическая криптография, теоретические проблемы информатики. Число научных публикаций — 68. alex.totrin@gmail.com; Приборостроительная, 35, Орел, 302034; р.т.: +7(486) 254-99-33.

Kozachok Alexander Vasilevich — Ph.D., researcher, The Academy of Federal Security Guard Service of the Russian Federation. Research interests: information security, unauthorized access protection, mathematical cryptography, theoretical problems of computer science. The number of publications — 68. alex.totrin@gmail.com; 35, Priborostroitelnaya Street, Orel, 302034, Russia; office phone: +7(486) 254-99-33.

Бочков Максим Вадимович — д-р техн. наук, заместитель директора по научной и учебной работе, частное образовательное учреждение дополнительного профессионального образования «Центр предпринимательских рисков» (ЧОУ ДПО "ЦПР"). Область научных интересов: информационная безопасность, защита от несанкционированного

доступа, защита информации в автоматизированных системах. Число научных публикаций — 150. mvboch@yandex.ru; ул. Профессора Попова, 27, Санкт-Петербург, 197022; р.т.: +7 (812) 234-95-66.

Bochkov Maksim Vadimovich — Ph.D., Dr. Sci., professor, deputy director for science, Business risk educational center. Research interests: information security, unauthorized access protection. The number of publications — 150. mvboch@yandex.ru; 27, Professor Popov Street, Saint-Petersburg, 197022, Russia; office phone: +7 (812) 234-95-66.

Фаткиева Роза Равильевна — к-т техн. наук, доцент, старший научный сотрудник лаборатории информационно-вычислительных систем и технологии программирования, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: моделирование информационных систем. Число научных публикаций — 50. rrf@iias.spb.su; 14-я линия В.О., д. 39, Санкт-Петербург, 199178; р.т.: +7-812-328-43-69, Факс: +7 (812)350-1113.

Fatkieva Roza Ravilievna — Ph.D., associate professor, senior researcher of computer and information systems and software engineering laboratory, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS). Research interests: modeling of information systems. The number of publications — 50. rrf@iias.spb.su; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7-812-328-43-69, Fax: +7 (812)350-1113.

Туан Лай Минь — сотрудник, Академия Федеральной службы охраны Российской Федерации. Область научных интересов: информационная безопасность, защита от несанкционированного доступа. Число научных публикаций — 8. lmtuan.1989@gmail.com; Приборостроительная, 35, Орел, 302034; р.т.: +7(486) 254-99-33.

Tuan Lai Minh — researcher, The Academy of Federal Security Guard Service of the Russian Federation. Research interests: information security, unauthorized access protection. The number of publications — 8. lmtuan.1989@gmail.com; 35, Priborostroitelnaya Street, Orel, 302034, Russia; office phone: +7(486) 254-99-33.

РЕФЕРАТ

Козачок А.В., Бочков М.В., Фаткиева Р.Р., Туан Л.М. **Аналитическая модель защиты файлов документальных форматов от несанкционированного доступа.**

Целью проводимого исследования является построение комплекса моделей и алгоритмов процесса контролируемого разграничения доступа к файлам документальных форматов, позволяющего осуществить защиту от несанкционированного доступа к информации за счет применения неразличимой обфускации программного кода.

В статье подробно рассмотрено описание аналитической модели неразличимой обфускации программного кода, положенной в основу предлагаемого подхода к защите от несанкционированного доступа к файлам документальных форматов.

Процедуру проверки прав доступа пользователя к документу, внедренному в контейнер, можно рассматривать как точечную функцию, поскольку результатом ее выполнения является значение из множества {0,1}, поэтому применение неразличимой обфускации для защиты данной проверки является также корректным. Данная работа посвящена модификации существующих подходов к осуществлению неразличимой обфускации с целью устранения ряда ограничений, обусловленных применяемыми механизмами, моделями и алгоритмами.

Разработанная модель неразличимой обфускации включает в себя пять основных этапов.

1. Преобразование булевой функций в вид ветвящейся программы.
2. Преобразование ветвящейся программы в вид матричной ветвящейся программы.
3. Рандомизация матричной ветвящейся программы.
4. Кодирование рандомизированной матричной ветвящейся программы с помощью схемы дифференциального кодирования.
5. Вычисление обфусцированной функции.

Отличительными особенностями разработанной модели являются:

— применение теоремы Сауэрхоффа для преобразования булевой функции в вид ветвящейся программы, что позволило добиться уменьшения длины формируемой ветвящейся программы;

— использование системы разделенных множеств для защиты от атак "смешенных произведений" и "смешанных входных данных";

— уточнение ряд процедур, осуществляемых на различных этапах неразличимой обфускации программного кода.

SUMMARY

Kozachok A.V., Bochkov M.V., Fatkieva R.R., Tuan L.M. **Analytical Model for Protecting Documentary File Formats from Unauthorized Access.**

The purpose of the research is to construct a complex of models and algorithms to control data access restriction to the documentary file formats based on indistinguishable program code obfuscation.

The article discussed in detail the description of the indistinguishable code obfuscation analytical model that underlies the proposed approach to protect against unauthorized access to the files of documentary formats.

The process of checking user access rights to the document, encapsulated in a container, may be considered as a point function because its value is from the set {0, 1}. So using indistinguishable obfuscation to protect this process is correct. This work is devoted to the modification of existing approaches to indistinguishable obfuscation and removing some restrictions, imposed by the applicable mechanisms, models and algorithms.

The developed model of indistinguishable obfuscation consists of five basic steps.

1. Convert Boolean function to branching program.
2. Convert branching program to matrix branching program.
3. Randomization of matrix branching program.
4. Encoding randomized matrix branching program with graded encoding scheme.
5. Evaluation of obfuscated function.

The distinctive features of the model are:

— the application of Sauerhoff Theorem to convert a Boolean function into the form of branching programs, which resulted in reducing the length formed by the branching program;

— the use of the divided sets to protect against attacks of mixed products and mixed input;

— clarifying a number of procedures carried out at different stages of indistinguishable code obfuscation.