УДК 004.93(015.7)

МАРКОВСКИЙ А.П.,
ЗЮЗЯ А.А.,
ШЕРСТЮК В.Д.

ПОЛУЧЕНИЕ БУЛЕВЫХ ПРЕОБРАЗОВАНИЙ СПЕЦИАЛЬНЫХ КЛАССОВ ДЛЯ ПОСТРОЕНИЯ ЭФФЕКТИВНЫХ АЛГОРИТМОВ ЗАЩИТЫ ИНФОРМАЦИИ

В статье представлены результаты исследований нелинейных булевых преобразований, обратные к которым обладают свойством неоднозначности и их использования в криптографических алгоритмах. Предложен новый метод получения булевых функциональных преобразований этого класса. Метод оперирует с процедурной формой представления булевых преобразований. Это позволяет строить преобразования от сотен булевых переменных. Булевы преобразования такого класса могут быть использованы для ускорения идентификации пользователей на основе концепции нулевых знаний. Установлены зависимости между временем построения преобразований и параметрами процедурной формы.

This paper presents investigation of nonlinear Boolean transformations inverse for which are ambiguous and its application in cryptographical algorithms. A new method for designing such class of Boolean transformations is suggested. The method deals with the procedure form representation of Boolean transformations. It allowed to buil Boolean transformatiom from hundreds Boolean variables. Boolean transformation on such class can be use for accelerate of user identification based on “zero-knoledge” conseption. The relationship between transformation building time and procedure form parameters is established.

Введение

Динамичное расширение и углубление процессов информационной интеграции требует адекватного совершенствования методов и средств защиты данных и контроля над правами доступа к ним.

К настоящему времени в основе большей часть систем защиты информации лежит использование криптографических механизмов. В свою очередь, такие механизмы базируются на математических преобразованиях, обладающих специфическими свойствами. Эти преобразования относятся к различным областям современной математики, но наиболее часто к теории чисел, конечных полей и булевой алгебре. Булевы преобразования обеспечивают существенно более высокую производительность реализации функций защиты информации в сравнении с другими преобразованиями.

Расширение использования средств защиты данных, в том числе в системах реального времени, требует радикального повышения скорости реализации вычислений, связанных с информационной безопасностью. Это определяет важность расширения возможностей использования для защиты информации именно булевых преобразований. Такое расширение может быть достигнуто путем получение преобразований со специальными свойствами, важными для защиты данных.

Таким образом, проблема расширения возможностей использования булевых функциональных преобразований в свете требований повышению скорости реализации функций защиты информации является актуальной.

Анализ проблемы получения необратимых преобразований, обратные к которым неоднозначны

Эффективность алгоритмов защиты информации определяется двумя факторами: уровнем защищенности и объемом вычислительных ресурсов, затрачиваемых на реализацию функций защиты.

В основе всех криптографических алгоритмов защиты информации лежит аналитически неразрешимая математическая задача [1]. В большинстве случаев практического использования, такие задачи имеют вид необратимого преобразования $Y=F(X)$, то есть преобразования, для которого определена алгоритмически функция $F(X)$ вычисления в прямом направлении и не существует аналитического способа получения функции $\Phi(Y)$ обратного преобразования $X=\Phi(Y)$ по известной функции $F(X)$. Единственным способом решения таких задач, то есть нахождения значения X для заданного значения Y при известной функции $F(X)$ является перебор значений X. Большая часть аналитически неразрешимых задач, ле-

жащих в основе современных криптографических алгоритмов относится к теории чисел, и булевой алгебре. В частности, к теории чисел относятся задачи дискретного логарифмирования, на основе которых построены большинство алгоритмов несимметричного шифрования (алгоритмов с открытым ключом), в том числе, широко используемые на практике RSA, El-Gamal, EEC, а также алгоритмы цифровой подписи, такие как DSS [1]. В основе достаточно широкого класса криптографических алгоритмов лежит аналитически неразрешимая задача булевой алгебры – отыскание корней систем нелинейных булевых уравнений. К этому классу алгоритмов относятся все алгоритмы блочного симметричного шифрования, такие как DES, IDEA, Rijndael, а также большая часть хеш-алгоритмов, в том числе, наиболее распространенные на практике RC-5, SHA и RIPEMD-160 [2].

Основным достоинством алгоритмов построенных на основе аналитически неразрешимых задач теории чисел является существование нескольких ключей. Так в алгоритмах шифрования RSA, El-Gamal, EEC, ключи, используемые для прямого и обратного преобразований различны. При наличии нескольких ключей, часть из них могут использоваться как открытые, а часть – как закрытые. Это позволяет строить на этой основе существенно более эффективные механизмы защиты информации и организации доступа к информационным ресурсам по сравнению с алгоритмами, имеющими единый ключ. Именно поэтому появление в 1978 г. алгоритма с “открытым” ключом – RSA, в основе которого лежит аналитически необратимое преобразование $F(X) = A^X \bmod M$, связывают с “открытием новой эры в технологии защиты информации” [2]. В теоретическом плане, существование нескольких ключей криптографического преобразования обусловлено тем, что необратимая задача является неоднозначной, то есть, существует, по крайней мере, два ключа X_1 и X_2 для которых выполняется $F(X_1) = F(X_2)$. Основным недостатком алгоритмов защиты информации, в основе которых лежат аналитически неразрешимые задачи теории чисел является низкая скорость их реализации, обусловленная высокой вычислительной сложностью операций модулярного экспоненцирования над числами, разрядность которых составляет тысячи.

Этого недостатка лишены алгоритмы защиты информации, в основе которых лежит аналитически неразрешимая задача булевой алгебры. Рекурсивное вычисление систем булевых функций может быть организовано достаточно эффективно как программными средствами на универсальных процессорах, так и специализированными аппаратными вычислителями. В оценочном плане, скорость криптографической обработки данных алгоритмами на основе булевых функций и на основе модулярной арифметики отличается на 3-4 порядка [3]. Однако, алгоритмы на основе булевых преобразований обладают существенно меньшими функциональными возможностями, которые не позволяют реализовать с их использованием эффективные протоколы защиты информации, подобно алгоритмам на основе неразрешимых задач теории чисел. В частности, использование булевых преобразований не позволяет реализовать базовые для современных технологий защиты информации концепции несимметричного шифрования данных, цифровой подписи сообщений, идентификации на основе схемы “нулевого знания”. Одним из факторов, обуславливающих ограниченные функциональные возможности алгоритмов на основе булевых преобразований является однозначность последних [3].

Одним из наиболее перспективных направлений совершенствования технологии защиты данных является расширение функциональных возможностей алгоритмов, основанных на булевой алгебре [2]. Это позволит строить на основе таких алгоритмов эффективные протоколы защиты данных в сетях, реализация которых требует существенно меньших вычислительных ресурсов и, соответственно, может выполняться на порядки быстрее, чем при использовании алгоритмов, основанных на модулярной арифметике.

В рамках реализации этого направления совершенствования технологии защиты информации важным представляется разработка методологии получения необратимых булевых преобразований, обладающих свойством неоднозначности [3]. Такие преобразования, в частности, могут быть эффективно использованы для создания эффективных протоколов идентификации удаленных абонентов много-

пользовательских систем на основе концепции “нулевых знаний” [4].

Целью исследований является разработка метода получения булевых функциональных преобразований, обладающих свойствами необратимости и неоднозначности обратного преобразования.

Метод получения преобразований специальных классов в процедурной форме

Для достижения поставленной цели необходимо сформировать булево функциональное преобразование $F(X)$, определенное на множестве 2^n значений n-битового бинарного вектора $X=\{x_1,x_2,\ldots,x_n\}$, $\forall i\in\{1,\ldots,n\}$: $x_i\in\{0,1\}$ результатом которого является n-битовый выходной бинарный вектор $Y=F(X)$, $Y=\{y_1,y_2,\ldots,y_n\}$, $\forall i\in\{1,\ldots,n\}$: $y_i\in\{0,1\}$. Каждую i-тую бинарную компоненту y_i выходного вектора Y можно рассматривать, как значение булевой функции $f_i(X)$, определенной на множестве значений X. Исходя из этого, можно говорить, что преобразование F(X) состоит из n булевых функций $f_1(X),f_2(X),\ldots,f_n(X)$: $F(X)=\{f_1(X),f_2(X),\ldots,f_n(X),\}$.

Для того, чтобы синтезируемое преобразование $F(X)$ обладало свойством необратимости, необходимо, чтобы каждая из булевых функций $f_1(X),f_2(X),\ldots,f_n(X)$, составляющих преобразование $F(X)$ должна быть нелинейной и зависеть от каждой из n входных переменных $X=\{x_1,x_2,\ldots,x_n\}$.

Для того, чтобы преобразование $F(X)$ обладало свойством неоднозначности, необходимо, чтобы существовало множество Ω входных векторов, на которых преобразование $F(X)$ принимает одинаковое значение: $\forall Q,G\in\Omega,\ Q\neq G$: $F(Q)=F(G)=U$.

Как известно, булевы функциональные преобразования могут быть представлены в следующих трех формах:

- в виде таблиц истинности;
- в алгебраических нормальных формах
- в процедурном виде, то есть в виде алгоритма вычисления выходного вектора Y по значению входного вектора X.

Применительно к задачам защиты информации, преимущественно, используется третья из указанных форм представления булевых функциональных преобразований, поскольку число переменных, на которых определены такие преобразования составляет сотни. Исходя из этого, предлагается формировать преобразование в процедурном виде.

В качестве базовой процедурной схемы вычисления преобразования $F(X)$ предлагается структура, показанная на рис.1.

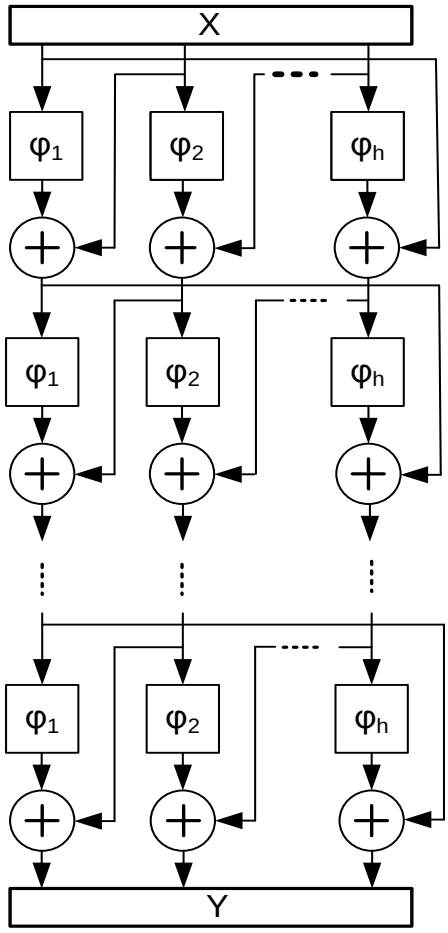

Рис.1. Структура процедурной формы вычисления булевых преобразований

Структура предполагает организацию вычисления $F(X)$ с разбиением данных на k-разрядные фрагменты. Если полагать, что n и k являются степенями 2, то число h фрагментов определяется как $h = n/k$. Подобно большинству структур, используемых для реализации булевых преобразований алгоритмов защиты информации, структура, показанная на рис.1, состоит из перемежающихся слоев нелинейного преобразования и перемешивания фрагментов. Нелинейное преобразование осуществляется с использованием h табличных булевых функциональных преобразований $\varphi_1,\varphi_2,\ldots,\varphi_h$. Каждое j-тое, $j\in\{1,\ldots,h\}$, фрагментарное преобразование $\varphi_j(V)$ определено на множестве 2^k всех возможных значений k-битового вектора $V=\{v_1,v_2,\ldots,v_k\}$, $\forall l\in\{1,\ldots,k\}$:

$v_l \in \{0,1\}$. Для перемешивания результатов обработки отдельных фрагментов используется суммирование по модулю 2 результатов нелинейных преобразований смежных фрагментов. Для того, чтобы каждый бит выходного вектора Y зависел от каждого из битов входного вектора X, число повторяющихся циклов в структуре преобразования должно быть не меньше h – числа фрагментов.

Если обозначить через Z_{qj} k-битовый вектор, являющийся j-тым фрагментом n–битового кода $W_q=\{Z_{q1},Z_{q2},\ldots,Z_{qh}\}$ формируемого на выходе q-того цикла, при этом $W_0 = X$, $W_h = Y$, то процедура вычисления функции, представленной на рис.1 аналитически может быть описана в виде:

$$\begin{aligned}
&\forall j \in \{1,...,h\} : Z_{0,j} = \{x_{(j-1)\cdot k+1}, x_{(j-1)\cdot k+2},..., x_{j\cdot k}\};\\
&\forall q \in \{1,...,h\}:\\
&\forall g \in \{1,..., g-1\} : Z_{qg} = \varphi_j(Z_{q-1,g}) \oplus Z_{q-1,g+1}\\
&Z_{qh} = \varphi_h(Z_{q-1,h}) \oplus Z_{q-1,1}
\end{aligned} \qquad (1)$$

Ниже через $W_q(Q)$ обозначено значение промежуточного кода на выходе q-го цикла для значения входного вектора $X=Q$. Соответственно, через $Z_{qj}(Q)$ обозначено код j-того фрагмента вектора $W_q(Q)$.

Задачей формирования неоднозначного и необратимого булевого преобразования $F(X)$ в рамках структуры, показанной на рис.1, является получение фрагментарных булевых функциональных преобразований – $\varphi_1,\varphi_2,\ldots\varphi_h$, которые для заранее выбранного множества Ω входных бинарных векторов $\Omega = \{X_1,X_2,\ldots,X_m\}$ обеспечивают единое значение результата, вычисляемого в соответствии с (2.10). Каждое j-тое, $j \in \{1,\ldots,h\}$, фрагментарное булево преобразование $\varphi_j(V)$ состоит из k булевых функций $\varphi_j = \{\phi_{j1}(V),\phi_{j2}(V),\ldots,\phi_{jk}(V)\}$ и может иметь результатом значения, принадлежащие интервалу от 0 до 2^k. Для решения этой задачи предлагается специальный метод, который состоит в последовательном определении значений фрагментарных функциональных преобразований в процессе преобразования входных векторов из множества Ω. Метод использует множество Θ, в которое включается для каждого из фрагментарных функциональных преобразований $\varphi_1,\varphi_2,\ldots\varphi_h$ список наборов их аргументов, на которых значение преобразования определяются случайным образом в процессе вычисления значения текущего входного вектора X_t.

Предлагаемый метод сводится к выполнению следующей последовательности действий:

1. Значения всех h фрагментарных преобразований $\varphi_1,\varphi_2,\ldots\varphi_h$ на всех 2^k возможных булевых наборах значений их аргументов принимаются равными –1, что соответствует неопределенному значению. Множество $\Omega=\varnothing$.
2. Выбирается произвольный n-битовый вектор X_1, который включается во множество Ω.
3. Вычисляется значение $U = F(X_1)$ в соответствием с (2.10). При этом, на каждом q–том из h слоев процедуры, для каждого j-го фрагмента (j=1,…h), если для предшествующего фрагмента $z_{q-1,j}(X_1)$ значение фрагментарного функционального преобразования не определено, то есть $\varphi_j(z_{q-1,j}(X_1))$=-1, то оно определяется как случайное число из интервала от 0 до 2^k-1.
4. Выбирается произвольно вектор X_t. Положить $W_0(X_t)=X_t$. Множество Θ полагается пустым: $\Theta=\varnothing$. Случайным образом генерируется номер N_c слоя ($N_c=\{1,\ldots,h\}$), на котором осуществляется уравнивание (“слияние”) промежуточных значений вектора X_t и одного из векторов множества Ω с порядковым номером меньшим t. Установить номер q текущего слоя в единицу: q=1. Если N_c=1, то переход на п.7.
5. Если q<N_c-1, то вычисление промежуточных значений для каждого j–го фрагмента осуществляется следующим образом: в случае $\varphi_j(z_{q-1,j})\neq$-1, то вычисление производится в соответствии с (1). Если значение фрагментарного функционального преобразования φ_j на наборе $z_{q-1,j}$ не определено: $\varphi_j(z_{q-1,j})$= -1, то случайным образом формируется принадлежащее интервалу от 0 до 2^k-1 значение φ_j на наборе $z_{q-1,j}$, после чего вычисление осуществляется согласно (1). При этом номер j доопределяемого преобразования φ_j вместе с набором $z_{q-1,j}$ заносятся в множество Θ: $\Theta=\Theta\cup <j,z_{q-1,j}>$. После вычисления таким образом всех h фрагментов промежуточных результатов q-слоя для кода X_t осуществляется переход к следующему слою, путем присваивания $q=q$+1 и возврата на повторное выполнение пп. 5.
6. Если $q=N_c$-1, и $\varphi_j(z_{q-1,j})$= -1, то значение преобразования φ_j на наборе $z_{q-1,j}$, выбирается таким образом, чтобы на наборе $z_{q,j} = \varphi_j(z_{q-1,j})\oplus z_{q-1,j+1}$ значение преобразования φ_j было не определено, то есть выполнялось: $\varphi_j(z_{q,j})$= -1. При этом номер j доопределяемого преобра-

зования φ_j вместе с набором $z_{q-1,j}$ заносятся в множество Θ: $\Theta=\Theta\cup <j,z_{q-1,j}>$. Если $\varphi_j(z_{q-1,j}) \neq -1$, то вычисление $z_{q,j}$ осуществляется в соответствии с (2.10). После вычисления таким образом всех h фрагментов промежуточных результатов q-слоя для кода X_t осуществляется переход к следующему слою, путем присваивания $q=q+1$.

7. Для каждого j-го фрагмента проверяется выполнение условие: $\varphi_j(z_{q-1,j}) = -1$, если это условие выполняется, то случайным образом $d\in\{1,\ldots,t-1\}$, после чего осуществляется переход на пп.8. Для каждого j-го фрагмента проверяется выполнение условие: $\varphi_j(z_{q-1,j})\neq -1$, но при этом существует такое $d\in\{1,\ldots,t-1\}$, что $\varphi_j(z_{q-1,j})\oplus z_{q-1,j+1} = z_{qj}(X_d)$. Если это условие выполняется, то осуществляется переход на пп.8. Если означенные выше условия не выполняются, то проведенный подбор значений фрагментарных функциональных преобразований для кода X_t является конфликтным. Все определенные при преобразовании X_t значения фрагментарных булевых преобразований вновь считаются неопределенными: $\forall <j,z>\in\Theta$: $\varphi_j(z)= -1$. Возврат на пп.4.

8. Для каждого j-го фрагмента выполняется следующее: если $\varphi_j(z_{q-1,j}) = -1$, то соответствующее значение фрагментарного преобразования на наборе определяется как: $\varphi_j(z_{q-1,j}(X_t)) = z_{q-1,j+1}(X_t) \oplus z_{q-1,j}(X_d)$, если $j<h$ и $\varphi_j(z_{q-1,j}(X_t)) = z_{q-1,1,}(X_t) \oplus z_{q-1,j+1}(X_d)$, если $j=h$.

9. Если $t < m$, то увеличивается на единицу номер t текущего вектора X из множества Ω: $t = t+1$. Возвращение на пп. 4.

10. Для всех h фрагментарных функциональных булевых преобразований $\varphi_1,\varphi_2,\ldots,\varphi_h$, на наборах, которые не определены раньше, установить значения, которые генерируется, как случайные целые числа, принадлежащие интервалу от 0 до 2^k–1: $\forall j\in\{1,\ldots,h\}$, $\forall z\in\{0,\ldots,2^k-1\}$: $\varphi_j(z)= -1$ определить : $\varphi_j(z)=$ Random$(0,2^k-1)$.

Результатом работы приведенной выше последовательности действий является h таблиц фрагментарных булевых преобразований $\varphi_1,\varphi_2,\ldots\varphi_h$, которые полностью задают, в совокупности со структурой, показанной на рис.1, процедуру вычисления преобразования $F(X)$.

Вероятность того, что результатом преобразования над случайным входным кодом X_e будет получено значение, совпадающее с U равно 2^{-n}. Учитывая, что численное значение n на практике составляет сотни, то вполне очевидно, что вероятность подбора идентифицирующего кода внешним по отношению к системе лицом, при использовании предложенного преобразования, практически равна нулю.

Использование разработанного метода получения нелинейных и неоднозначных булевых преобразований иллюстрируется следующим примером. Пусть n – разрядность кодов на входе и выходе преобразования $F(X)$ равна 16, то есть n=16. Пусть, далее, обрабатываемые преобразованием $F(X)$ коды разделяются на 4 фрагмента (h=4), каждый по 4 разряда ($k=n/h$=4). Это означает, что процедура преобразования состоит из 4-х слоев. Фрагментарные табличные булевы преобразования $\varphi_1,\varphi_2,\varphi_3,\varphi_4$ определены на множестве 16-ти возможных значений 4-х булевых переменных и каждое из низ содержит 4 булевых функции, так, что каждое из упомянутых преобразований может принимать значения от 0 до 15-ти. Пусть множество Ω состоит из 3-х 16-битовых ключей, которые могут быть представлены с использованием 16-ричной системы в следующем виде: X_1=5E96h, X_2 = B1C6h, X_3 = 7A48h. Динамика заполнения таблиц истинности фрагментарных булевых функциональных преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$ в процессе обработки ключей, принадлежащих множеству Ω отражена в таблице 1. Незаполненные клетки таблицы соответствуют неопределенному значению фрагментарных булевых преобразований на соответствующих наборах входных переменных.

Результатом работы предложенного метода являются таблицы истинности фрагментарных булевых преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$.

После обработки всех 3-х кодов X_1,X_2,X_3 в значения фрагментарных функциональных преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$ определены на 64% наборов значений их входных переменных. Соответственно, на 36% наборов в соответствии с пп.10 изложенного выше метода, значения функций определяются случайным образом.

Табл.1. Пример заполнения таблиц фрагментарных преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$

X	Таблицы значений фрагментарных булевых преобразований							
	После обработки X_1				В окончательном виде			
	φ_1	φ_2	φ_3	φ_4	φ_1	φ_2	φ_3	φ_4
0			11		12	4	11	8
1					3	14	9	2
2					0	0	6	7
3	4				4	8	4	11
4	8				8	12	6	5
5	10				10	7	3	8
6				2	3	10	7	2
7			12	15	13	15	12	15
8					12	7	15	6
9	3		6		3	9	6	8
10		15			5	15	15	10
11		13		14	15	13	10	14
12			12		8	9	12	11
13		6		0	5	6	10	0
14		2			8	2	12	9
15					0	3	15	3

Формирование фрагментарных булевых функциональных преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$ в рамках изложенного метода выполняется с использованием проб: если для при обработке очередного входного вектора $X\in\Omega$, в процессе которого используется случайное определение значений преобразований $\varphi_1,\varphi_2,\varphi_3,\varphi_4$ на определенных наборах, не удается избежать конфликта с ранее определенными значениями, то, согласно пп.7 осуществляется возврат на повторную обработку вектора X. Таким образом, основным фактором, от которого зависит время формирования нелинейного функционального преобразования является количество проб заполнения таблиц фрагментарных функциональных преобразований.
при обработке вектора окажется успешной определяется следующим выражением:

$$P_g = (1-\frac{h^2\cdot g^2}{2^{2\cdot(k+1)}})^h \qquad (2)$$

Соответственно, среднее число t_g проб, затрачиваемых для заполнения таблиц фрагментарных преобразований при обработке g–го входного вектора определяется как: $t_g = 1/P_g$. Тогда, среднее число T_m проб, затрачиваемых для заполнения таблиц фрагментарных преобразований при обработке всех m входных кодов, составляющих множество Ω, определяется как сумма среднего числа проб, требующихся для обработки каждого из m входных кодов:

$$T_m = \sum_{j=1}^{m}\frac{1}{(1-\frac{h^2\cdot j^2}{2^{2\cdot(k+1)}})^h} \qquad (3)$$

Общий объем V_T памяти таблиц фрагментарных нелинейных булевых преобразований определяется в виде:

Исходя из сказанного, важным аспектом анализа предложенного метода получения неоднозначного и необратимого функционального булевого преобразования в процедурной форме является оценка зависимости числа проб от ее параметров.

Пусть, обрабатывается g-тый по очереди входной вектор X_g множества Ω. Поскольку при обработке вектора, в среднем, определяется значения фрагментарных преобразований на половине слоев, то среднее число значений одного преобразования, определяемых при обработке одного вектора X составляет $h/2$. Из этого следует, что к концу обработки g–го входного вектора X_g в каждом из h фрагментарных преобразований будет определено, в среднем, $h\cdot g/2$ значений. Из приведенного выше описания метода (пп.7) следует, что конфликтная ситуация возникает, если при обработке j-го фрагмента $z_{q,j}$ промежуточного значения на q-том слое значения соответствующего преобразования φ_j ранее определены как на наборе $z_{q,j}$, так и на наборе $\varphi_j(z_{q,j})\oplus z_{q-1,j+1}$ для $j<h$ или наборе $\varphi_j(z_{q,j})\oplus z_{q-1,1}$ для $j=h$. То есть, для того, чтобы конфликтная ситуация возникла, необходимо, чтобы для двух фиксированных наборов значения функционального преобразования φ_j были ранее определены. Учитывая, что общее число наборов, на которых определяется φ_j равно 2^k, из них, в среднем, на $h\cdot g/2$ значение преобразования φ_j уже определено ранее, то вероятность того, что на двух фиксированных наборах значение φ_j определено ранее составляет $(h\cdot g)^2/2^{2\cdot(k+1)}$. Принимая во внимание, что для повторной обработки вектора достаточно наличие конфликтной ситуации в одном фрагменте, вероятность P_g того, что проба случайного заполнения всех h фрагментарных преобразований $\varphi_1,\varphi_2,\ldots,\varphi_h$

$$V_T = h \cdot 2^{\frac{n}{h}} \quad (4)$$

Для проверки полученной теоретическим путем зависимости времени формирования функционального преобразования в зависимости от параметров процедурной формы и числа m возможных входных векторов были проведены статистические исследования с применением программной модели. Результаты экспериментальных исследований показали, что теоретическая оценка (3) упомянутых зависимостей, в целом, соответствует результатам, полученным при статистическом моделировании. Так, при длине кода доступа n=256 при параметрах процедурной формы h=16 и k=16, ее построение для числа кодов доступа m=4000 требует около 40000 проб, что занимает не более часа работы персонального компьютера. При этом множество Ω содержит около 4000 входных векторов, для которых результат преобразования $F(X)$ одинаков. При этом в процессе построения функционального преобразования заполнено около 47% таблиц фрагментарных преобразований. Остальные наборы таблиц в соответствии с пп.10 заполняются случайным образом.

Выводы

В результате проведенных исследований предложен метод получения процедурной формы нелинейных булевых функциональных преобразований, обратное к которым обладает свойством неоднозначности. Такие преобразования позволяют использовать их в некоторых применениях вместо модулярных операций над большими числами, что обеспечивает значительное (на 2-3 порядка) повышение производительности.

Установлены зависимости между параметрами процедурной формы, временем реализации алгоритма и характеристиками булевых перобразований.

Разработанный метод позволяет получать булевы функциональные преобразования, применение которых обеспечивает повышение скорости реализации алгоритмов защиты информации и средств идентификации удаленных абонентов.

Список литературы

1. Иванов М.А., Криптографические методы защиты информации в компьютерных системах и сетях. М.: ”Кудиз-образ”, 2001.– 368 с.
2. Самофалов К.Г., Марковский А.П., Гаваагийн Улзисайхан, Бардис Н., Метод получения булевых балансных SAC-функций для систем защиты информации. // Вісник Національного технічного університету України ”КПІ”. Інформатика, управління та обчислювальна техніка.–1998.– № 31.– С.131-140.
3. Seberry J., Zhang X., Zheng Y. Nonlinearity and propagation characteristics of balanced Boolean functions.//Information and Computation Academic Press. 1995.–Vol. 119, № 1 –P.1-13.