АЛГОРИТМ АСИММЕТРИЧНОГО ШИФРОВАНИЯ НА ОСНОВЕ БИНАРНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Е.И. Коновалова, А.М.Саяпин

Самарский государственный аэрокосмический университет им. академика С.П. Королёва (национальный исследовательский университет),

В настоящей работе предлагается алгоритм асимметричного шифрования, основанный на графе, вершинами которого являются двоичные последовательности (последовательность 0 и 1). Алгоритм является альтернативой существующим алгоритмам.

В 2005 году В.И. Арнольд предложил теорию об определении сложности конечных последовательностей, основанной на представлении множества последовательностей в виде графа.

Пусть x - последовательность из 0 и 1: $x=\left(x_1,x_2,...,x_n\right),\ x_i=\{0,1\}$ Множество M всех таких последовательностей конечно, его мощность равна, 2^n. Определим оператор взятия разности $A{:}M\to M,\ Ax=y$, где $x,\ y$ - бинарные последовательности длины n, по следующему правилу: $A\left(x_1,x_2,...,x_n\right)=\left(|x_2-x_1|,...,|x_n-x_1|\right)$.

Все бинарные последовательности длины n можно представить в виде ориентированного графа, вершины которого есть все 2^n элементов множества M. При этом существует единственная дуга из последовательности x в y если и только если $Ax=y$.

На рисунке 1 представлена часть графа для n=6. На рисунке представлены две компоненты связности, два цикла длины 1 и 6. Всего для n=6 имеется 4 компоненты связности, 4 цикла, два из которых имеют длину 6, один длины 3 и один длины 1. Каждой вершине, принадлежащей циклу, соответствует бинарное дерево высоты 2

В общем случае, длина цикла может быть любой, а размер двоичного дерева зависит от n и вычисляется по формуле $2^{g(n)}$, где функция $g(n)$ возвращает максимальную степень числа 2, входящую в разложение n. В дальнейшем будем считать p=g(n). В работе [2] приведена таблица, описывающая для $n\le 300$ структуру графа двоичных последовательностей

Следующая теорема, доказанная в [3], помогает эффективно проверить, принадлежит ли последовательность циклу или нет.

Теорема. Пусть $f(a)=\tilde{a}_1\ XOR\ \tilde{a}_2\ XOR...XOR\ \tilde{a}_{m-1}\ XOR\tilde{a}_m$, где $\tilde{a}_i$ подпоследовательность a длины p, $m=\dfrac{n}{p}$, тогда $f(a)=0$ $\Leftrightarrow$ последовательность принадлежит циклу.

На рисунке 1 выберем, например, последовательность a=110011, n=6, p=2, тогда значение хеш-фунцкии, определенной в теореме, равно f(a)=11 XOR 00 XOR 11 =0, значит, последовательность принадлежит циклу. Вообще говоря, значение хеш-функции показывает положение последовательности в графе. На рисунке 1 значение f(a) указано в квадратных скобках.

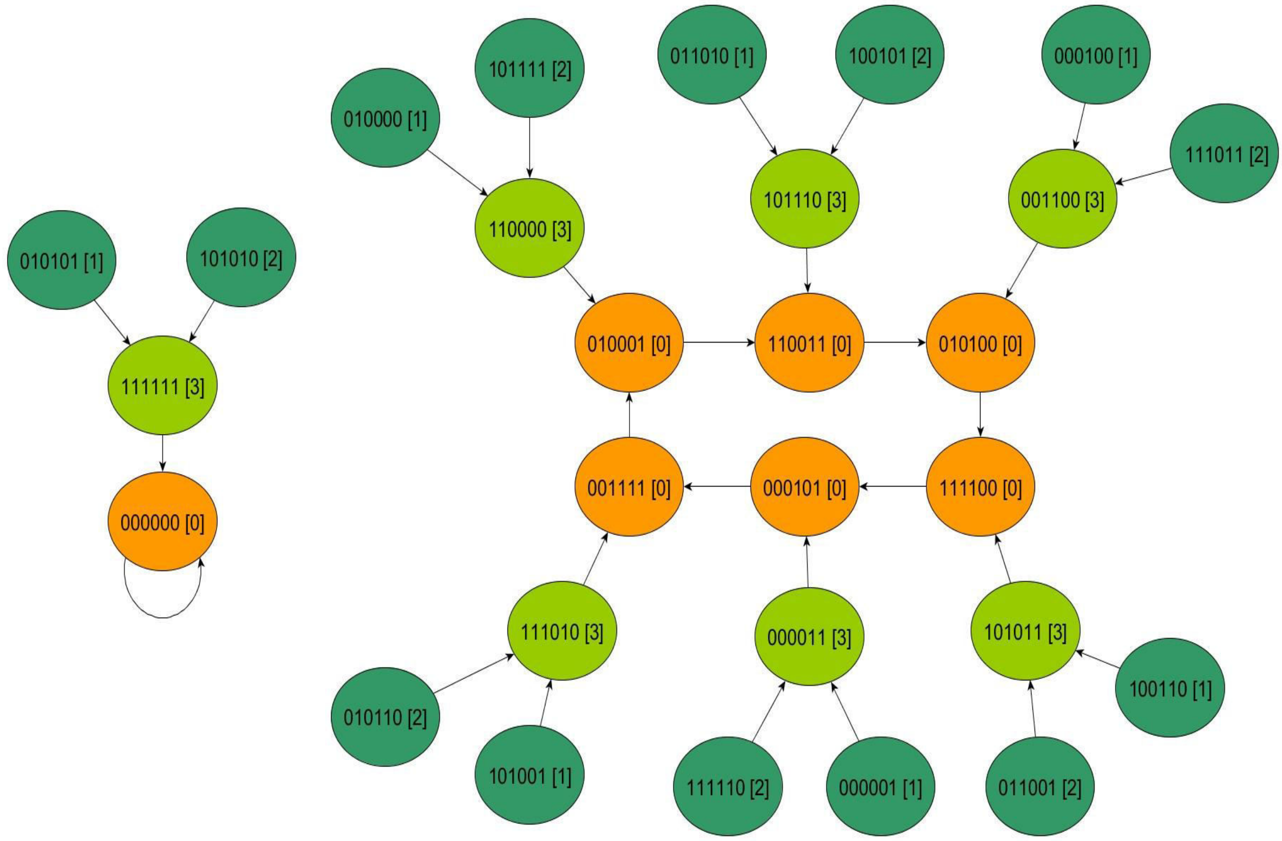

Рисунок 1 – Часть графа для n=6

Наряду с оператором A мы будем также использовать оператор сдвига G и оператор прыжка H. Определим оператор сдвига G$:M \to M$, $Gx = y$, где x, y - двоичные последовательности длины n, по следующему правилу: $A(x_1, x_2, \ldots, x_n) = (x_2, x_3, \ldots, x_1)$. Оператор прыжка H$:M \to M$, $H(x) = y$, где x, y - двоичные последовательности длины n, по следующему правилу: $H_k(x) = x\, XOR\, A(x)\, XOR\, A^2(x)\, XOR \ldots . XOR\, A^k(x)$, где k натуральное число.

Операторы G и H позволяют попадать в разные компоненты графа. Например, G(001111)=011110 (не представлена на рисунке), H_5(001111)=000000. Это свойство операторов повышает криптостойкость алгоритма шифрования.

Заметим, что операторы A, G, и H являются коммутирующими, это свойство операторов является определяющим свойством при построении алгоритма.

В основу криптографической системы положено предположение, о том, что действие оператора A можно считать односторонней функцией. Под односторонностью понимается не теоретическая однонаправленность, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства, за разумный интервал времени.

Авторами предлагается следующий алгоритм асимметричного шифрования.

Создание открытого и закрытого ключа:

1 Выбирается длина последовательности n.

2 Выбирается m произвольных значений $m_1, m_2, \ldots, m_k$.

3 Вычисляется значение $z = A^{m_1}(a_1) XORA^{m_2}(a_2) XOR \ldots XORA^{m_k}(a_k)$.

4 Открытым ключом будет последовательность образующих $a_1, a_2, \ldots, a_k$ и контрольная сумма z, закрытым ключом количество шагов для каждой образующей $m_1, m_2, \ldots, m_k$.

Верификация открытого ключа:

1 Произвольным образом выбираются числа α,β,γ,δ

2 Выполняются преобразования образующих: $b_1=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_1)$, $b_2=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_2)$, $\ldots b_k=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_k)$.

3 Новые образующие $b_1, b_2,..., b_k$ посылаются владельцу открытого ключа.

4 Владелец открытого ключа выполняет преобразования образующих и отправляет для верификации: $Z' = A^{m_1}(b_1) xor A^{m_2}(b_2) xor ... xor A^{m_k}(b_k)$.

5 Проверка подлинности владельца открытого ключа заключается в проверке равенства: $A^{\alpha}G^{\beta}(H_{\gamma})^{\delta}(Z) = Z'$.

Рассмотрим вычислительные аспекты этого алгоритма. Во-первых, заметим, что сложность самого вычисления открытого ключа линейна. Всего нам потребуется $\sum_{i=1}^{k} m_i$ применений оператора A. Во-вторых, злоумышленник для того чтобы понять, какой приватный ключ соответствует открытому, должен решить уравнение $z = A^{m_1}(a_1) XORA^{m_2}(a_2) XOR ... XORA^{m_k}(a_k)$ относительно m_1, $m_2,\ldots m_k$. Тогда общее количество операций для такого перебора будет $k^{\sum_{i=1}^{k} m_i}$, что является сложной вычислительной задачей с экспоненциальным временем работы. Другая стратегия злоумышленника заключается в подборе преобразований $A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}$ таких, чтобы выполнялись уравнения $b_1=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_1)$, $b_2=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_2)$, $\ldots b_k=A^{\alpha}G^{\beta(}H_{\gamma})^{\delta}(a_k)$. Сложность решения этой задачи можно оценить как $O(r)$, с $O(r)$ дополнительной памятью, где r – размер наибольшего цикла в графе.

Параметры в алгоритме следует брать исходя из возможностей используемого ЭВМ.

Ниже приведены рекомендации для персонального компьютера. Для суперкомпьютеров эти параметры будут отличаться большей размерностью.

Рекомендуемые параметры для алгоритма:

- n = {293,283,281,271,263}
- k= [50..100]
- α = [100..1000]; β =[1..n]; γ = [100..1000]; δ = [1..1000];

Литература

1. В.И. Арнольд, Лекция: Сложность конечных последовательностей нулей и единиц и геометрия конечных функциональных пространств, 13.05.2006г., БКЗ Академический РАН, http://elementy.ru/lib/430178/430281

2. Lerner E.Yu. Tables of graphs of binary and ternary sequences differentiation.Preprint http://arxiv.org/abs/0704.2947v1.

3. Саяпин А.М. Сложность бинарных последовательностей// Вестник СМУиС – Самара: СГАУ, 2013. С82-85.