Применение нестандартных операций в алгоритмах симметричного и ассиметричного шифрования

Анастасия Годнева

МГУ им. М.В.Ломоносова

15 апреля 2014

Введение

В основе исследования - алгоритмы, использующиеся в стандартах шифрования республики Узбекистан.

В алгоритме формированя и проверки электронной подписи в роли односторонней функции выступает возведение в степень с параметром.

Алгоритм шифрования данных - симметричный алгоритм, в котором в качестве раундовых операций используются преобразования, зависящие от ключей.

В работе исследуется целесообразность использования этих операций и проводится сравнение стандартов шифрования с известными и считающимися надежными алгоритмами.

Умножение с параметром

Пусть $n \in \mathbb{N}$, $a, b, R \in \mathbb{Z}_n$. Тогда результат умножения элементов a и b с параметром R обозначается $a \circledR b$ и определяется следующим образом:

$$a \circledR b = a + b + abR \pmod{n}.$$

Для любого R умножение с параметром является коммутативной и ассоциативной операцией, с нейтральным элементом 0.

Умножение с параметром

Пусть $n \in \mathbb{N}$, $a, b, R \in \mathbb{Z}_n$. Тогда результат умножения элементов a и b с параметром R обозначается $a\circledR b$ и определяется следующим образом:

$$a\circledR b = a + b + abR \pmod{n}.$$

Для любого R умножение с параметром является коммутативной и ассоциативной операцией, с нейтральным элементом 0.

$a^{\backslash t}$ — результат возведения a в степень t относительно введенной операции.

Умножение с параметром

Пусть $n \in \mathbb{N}$, $a, b, R \in \mathbb{Z}_n$. Тогда результат умножения элементов a и b с параметром R обозначается $a \circledR b$ и определяется следующим образом:

$$a \circledR b = a + b + abR \pmod{n}.$$

Для любого R умножение с параметром является коммутативной и ассоциативной операцией, с нейтральным элементом 0.

$a^{\backslash t}$ — результат возведения a в степень t относительно введенной операции.

Элемент a из Z_n обратим по умножению с параметром R тогда и только тогда, когда $(1 + Ra)$ взаимно просто с n

Алгоритм создания электронной цифровой подписи

При формировании ЭЦП отправитель прежде всего вычисляет хэш-функцию $h(M)$ подписываемого текста M. Затем число $h(M)$ шифруется секретным ключом отправителя с помощью односторонней функции. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

В качестве односторонней функции АЭЦП использует использует возведение в степень с параметром, и в стандарте указано, что вычисления по этой операции осуществляются на том же уровне трудоемкости, что и в операциях возведения в степень, а инвертирование (обращение) функции требует не меньших вычислительных затрат и времени, чем в процессе решения проблемы дискретного логарифма. На этом последнем свойстве и основана стойкость алгоритма.
Более подробно описание АЭЦП приведено в [1].

Постановка задачи

Основные свойства умножения с параметром, представляющие интерес для его применения - это общие свойства группы обратимых элементов, существование элементов, которые можно возвести в большую степень и сложность взятия дискретного логарифма.
Случай, когда R и n взаимно просты был подробно рассмотрен в работе [2]. Была получена формула для возведения в степень с параметром обратимого элемента a:

$$a^{\backslash t} = \frac{(1 + R * a)^t - 1}{R} \, (mod\ n). \tag{1}$$

Задача дискретного логарифмирования в этом частном случае сводится к аналогичном задачам в группе обратимых элементов Z_n^*. Поэтому основной акцент будет делаться на на случае, когда $gcd(n, R) \neq 1$

Структура группы обратимых элементов

Пусть $n \in \mathbb{N}$ и $0 \leq R \leq n-1$. Обозначим через $\mathbb{Z}(n, R)$ группу элементов, обратимых относительно умножения с параметром R и считаем, что $R \neq 0$.

Теорема

Пусть $n = \prod_{k=1}^{q} p_k^{\alpha_k} * n_0$, $R = \prod_{k=1}^{q} p_k^{\beta_k} * r_0$, *и* $R < n$, $gcd(r_0, n) = 1$, $gcd(n_o, R) = 1$, *где* p_k — *все простые множители, общие для* R *и* n, *в порядке возрастания. Тогда группа* $\mathbb{Z}(n, R)$ *обратимых элементов следующим образом может быть разложена в произведение групп, каждая из которых или является циклической, или известным образом раскладывается в произведение циклических :*

$$\mathbb{Z}(n, R) \cong \mathbb{Z}^*_{n_0} \times \mathbb{Z}_2 \times \mathbb{Z}_{2^{\alpha_1 - 1}} \times \mathbb{Z}_{p_2^{\alpha_2}} \times \ldots \times \mathbb{Z}_{p_q^{\alpha_q}} \text{ при } p_1 = 2 \text{ и } \beta_1 = 1;$$

$$\mathbb{Z}(n, R) \cong \mathbb{Z}^*_{n_0} \times \mathbb{Z}_{p_1^{\alpha_1}} \times \mathbb{Z}_{p_2^{\alpha_2}} \times \ldots \times \mathbb{Z}_{p_q^{\alpha_q}} \text{ в остальных случаях.}$$

Образующие элементы подгрупп разложения вида $Z_{p_i^{\alpha_i}}$ *могут быть найдены по формуле* $n'_i = n_0 * \ldots * n_{i-1} * n_{i+1} * \ldots * n_q$, *где* $n_i = p_k^{\alpha_k}$

Алгоритм дискретного логарифмирования при условии известного разложения n и R

Под дискретным логарифмированием относительно параметра R и модуля n будем понимать решение уравнения:
$a^{\backslash x} = b \pmod{n}$.
Обозначим через $L(n, R)$ и $M(n, R)$ временную и пространственную сложность решения задачи дискретного логарифмирования в группе $\mathbb{Z}(n, R)$. Временной сложностью будем считать количество арифметических операций — сложения, вычитания, умножения, деления. Прежде, чем перейти к общему случаю, рассмотрим группу, в которой число n является степенью простого числа p, причем p делит R. В этом случае:

Теорема

Пусть $n = p^{\alpha}$, $R = r * p^{\beta}$, $gcd(r, p) = 1$, $\alpha, \beta \in \mathbb{N}$. *Тогда* $L(n, R) \leq c * k^3$, $M(n, R) \leq 3k * p^{\beta}$, *где* $k = \lceil \frac{\alpha}{\beta} \rceil$.

Алгоритм дискретного логарифмирования при условии известного разложения n и R

Следствие

В в случае условий теоремы 2 сложность решения задачи дискретного логарифмирования может быть оценена сверху слеующим образом:

$$L(n, R) \leq c * (log_p n)^3,$$

то есть задача полиномиальна.

Алгоритм дискретного логарифмирования при условии известного разложения n и R

В общем случае:

Теорема

Пусть $n = \prod_{k=1}^{q} p_k^{\alpha_k} * n_0$, $R = \prod_{m=1}^{q} p_m^{\beta_m} * r_0$, *как и в теореме 1. Тогда задача дискретного логарифмирования может быть разложена по взаимно-простым множителям числа n* $n_i = p_i^{\alpha_i}$ *и решаться отдельно по модулю каждого из этих множителей. При этом сложность задачи дискретного логарифмирования может быть оценена следующей величиной*

$$L(n, R) = L(n_0) + c * \sum_{k=1}^{q} \lceil \frac{\alpha_k}{\beta_k} \rceil^3 + q^2.$$

Упрощение задачи факторизации

Для решения задачи дискретного логарифмирования требуется представить n и R в виде произведения простых сомножителей, то есть решить задачу факторизации, имеющей субэкспоненциальную сложность решения . Но эта задача упростится в случае, когда нужно всего лишь найти общие для n и R делители.
Для натурального M обозначим через $F(M)$ сложность разложения M в произведение простых сомножителей.

Теорема

Для того, чтобы представить числа n и R в виде $n = \prod_{k=1}^{q} p_k^{\alpha_k} * n_0$, $R = \prod_{m=1}^{q} p_m^{\beta_m} * r_0$, *существует алгоритм, сложность которого не превышает* $\sum_{i=1}^{q} \alpha_i + F(M)$, *где* $M = \prod_{i=1}^{q} p^{gcd(\alpha_i, \beta_i)}$.

АШД - основные свойства

Алгоритм шифрования данных основан на использовании преобразований перемешивания столбцов и по полубайтовой (по байтовой) замены и характеризуется следующими признаками:

- Ключи этапа формируются на основе ключа шифрования и периодически обновляемого функционального ключа
- В перемешивании столбцов участвуют секретные параметры
- В преобразованиях по полубайтовой или по байтовой замене принимают участие элементы ключа
- Криптографические преобразования выполняются со значением $p = 16$, если длина входного блока и ключа шифрования равна 128 битам

АШД - основные свойства

Алгоритм шифрования данных основан на использовании преобразований перемешивания столбцов и по полубайтовой (по байтовой) замены и характеризуется следующими признаками:

- Ключи этапа формируются на основе ключа шифрования и периодически обновляемого функционального ключа
- В перемешивании столбцов участвуют секретные параметры
- В преобразованиях по полубайтовой или по байтовой замене принимают участие элементы ключа
- Криптографические преобразования выполняются со значением $p = 16$, если длина входного блока и ключа шифрования равна 128 битам

Согласно описанию стандарта перечисленные признаки обеспечивают повышение криптостойкости алгоритма шифрования данных (более подробно описано в [6]).

АШД - схема алгоритма (для длины входного блока 128 бит)

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;
5. $K_s \times H$ умножение на матрицу сеансового ключа слева (по половинам);

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;
5. $K_s \times H$ умножение на матрицу сеансового ключа слева (по половинам);
6. Циклический сдвиг $K_s e$ и H (H сдвигается по строчкам на разное количество полубайт);

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_se = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит k_se записываются в матрицу $K_s[4, 8]$;
3. Из k_se формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;
5. $K_s \times H$ умножение на матрицу сеансового ключа слева (по половинам);
6. Циклический сдвиг K_se и H (H сдвигается по строчкам на разное количество полубайт);
7. Операция по полубайтовой замены - S-блок. $S(H[i, j] = (H[i, j]^{-1} \pmod{17})(1 + K_1[i, j] * K_2[i, j] + K_2[i, j] \pmod{16}$;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;
5. $K_s \times H$ умножение на матрицу сеансового ключа слева (по половинам);
6. Циклический сдвиг $K_s e$ и H (H сдвигается по строчкам на разное количество полубайт);
7. Операция по полубайтовой замены - S-блок. $S(H[i, j] = (H[i, j]^{-1} \pmod{17})(1 + K_1[i, j] * K_2[i, j] + K_2[i, j] \pmod{16}$;
8. Формирование этапных ключей из $k_s e$;

АШД - схема алгоритма (для длины входного блока 128 бит)

1. Входной блок записывается в виде матрицы полубайтов $H[4, 8]$;
2. Сеансовый ключ формируется следующим образом $k_s e = k + k' * (1 + k_f * k)$ (336 бит) и первые 80 бит $k_s e$ записываются в матрицу $K_s[4, 8]$;
3. Из $k_s e$ формируются 2 массива этапных ключей $K_1[4, 8]$ и $K_2[4, 8]$.;
4. $H \oplus K_1$ - сложение по модулю 2 с этапным ключом;
5. $K_s \times H$ умножение на матрицу сеансового ключа слева (по половинам);
6. Циклический сдвиг $K_s e$ и H (H сдвигается по строчкам на разное количество полубайт);
7. Операция по полубайтовой замены - S-блок. $S(H[i, j] = (H[i, j]^{-1} \pmod{17})(1 + K_1[i, j] * K_2[i, j] + K_2[i, j] \pmod{16}$;
8. Формирование этапных ключей из $k_s e$;
9. Повторить 6-8 раз, начиная с пункта 4;

Функция рассеивания АШД

В роли функции рассеивания в АШД используется умножение слева на матрицу, состоящую из элементов сеансового ключа. Функция рассеивания должна удовлетворять следующим свойствам:

- Обратимость - достигается приведением матрицы $K_s[i, j]$ к специальной структуре, позволяющей легко проверить обратимость
- Линейность - позволяет применить алгоритм прямого расшифрования
- Симметричность - то есть функция должна работать со всеми данными единообразно
- Сильное рассеивание - изменение 1 полубайта исходного блока должно распространится на максимальное количество полубайт шифротекста, то есть на максимальное количество S-блоков.

Второе свойство делает шифр менее уязвимым к дифференциальному криптоанализу. Так как умножение на матрицу слева действует независимо на столбцы исходной матрицы, то имеет смысл рассматривать коэффициент распространения линейного преобразования для столбцов.

Коэффициент распространения функции рассеивания АШД

Коэффициентом распространения линейного преобразования функции F для столбца a называется следующая величина:
$d = \min_{a \neq 0}(W(a) + W(F(a))$
где $W(a)$ - байтовый вес или количество ненулевых полубайтов в столбце.
Легко проверяется следующее утверждение:

Утверждение

Если входные данные представляют собой массив полубайтов [4x4], а функция рассеивания - умножение на квадратную матрицу K_s слева, то $d(K_s) \leq \min 0 \leq i \leq 31 + n_i(K_s)$, где $n_i(K_s)$ - количество элементов в i-м столбце матрицы K_s обратимых по умножению в модуле $p = 16$

Коэффициент распространения функции рассеивания АШД

Следствием предыдущего утверждения является то, что существует целый класс ключей, допускающих функцию рассеивания с коэфициентом распространения равным 2. Это все ключи, которые, записанные в виде матрицы, содержат столбец с одним нечетным и 3мя четными элементами. Проверка, отсеивающая такие ключи в стандарте отсутствует. Возможны такие варианты исправления этого:

Коэффициент распространения функции рассеивания АШД

Следствием предыдущего утверждения является то, что существует целый класс ключей, допускающих функцию рассеивания с коэфициентом распространения равным 2. Это все ключи, которые, записанные в виде матрицы, содержат столбец с одним нечетным и 3мя четными элементами. Проверка, отсеивающая такие ключи в стандарте отсутствует. Возможны такие варианты исправления этого:

- Проверить, чтобы хотя бы 2 элемента в каждом столбце K_s были нечетными, иначе сделать замену для некоторых [i,j] $K[i,j] := K[i,j] + 1$ -> $d(K_s) \geq 3$

Коэффициент распространения функции рассеивания АШД

Следствием предыдущего утверждения является то, что существует целый класс ключей, допускающих функцию рассеивания с коэфициентом распространения равным 2. Это все ключи, которые, записанные в виде матрицы, содержат столбец с одним нечетным и 3мя четными элементами. Проверка, отсеивающая такие ключи в стандарте отсутствует. Возможны такие варианты исправления этого:

- Проверить, чтобы хотя бы 2 элемента в каждом столбце K_s были нечетными, иначе сделать замену для некоторых [i,j] $K[i,j] := K[i,j] + 1$ -> $d(K_s) \geq 3$
- Проверить, чтобы хотя бы 3 элемента в каждом столбце были нечетными и каждый минор 3x3 обратим - $D(K_s) \geq 4$

Коэффициент распространения функции рассеивания АШД

Следствием предыдущего утверждения является то, что существует целый класс ключей, допускающих функцию рассеивания с коэфициентом распространения равным 2. Это все ключи, которые, записанные в виде матрицы, содержат столбец с одним нечетным и 3мя четными элементами. Проверка, отсеивающая такие ключи в стандарте отсутствует. Возможны такие варианты исправления этого:

- Проверить, чтобы хотя бы 2 элемента в каждом столбце K_s были нечетными, иначе сделать замену для некоторых [i,j] $K[i,j] := K[i,j] + 1$ -> $d(K_s) \geq 3$
- Проверить, чтобы хотя бы 3 элемента в каждом столбце были нечетными и каждый минор 3x3 обратим - $D(K_s) \geq 4$
- Если делать все элементы нечетными - матрица перестанет быть необратимой, то есть нельзя добиться, чтобы $d = 5$ всегда.

Функция перемешивания АШД

В качестве функции перемешивания алгоритма шифрования данных используется следующая функция по полубайтовой замены, зависящая от раундовых ключей:
$S(H[i,j] = (H[i,j]^{-1} \pmod{17})(1 + K_1[i,j] * K_2[i,j]) + K_2[i,j] \pmod{16}$
Она является единственной нелинейной операцией в алгоритме и должна удовлетворять следующим условиям:

- Обратимость, т.е. $\nexists h_1 \neq h_2 : S(h_1) = S(h_2)$
- S-блок не должен быть с большой вероятностью приближен линейным соотношением

Первое свойство достигается проверкой того, что $K_1[i,j] * K_2[i,j] \equiv 0 \pmod{2}$, со вторым свойством обстоит хуже.

Линейное приближение функции перемешивания АШД

Цель линейного приближения функции перемешивания - это нахождение корреляции между входом и выходом функции, которую потом можно попытаться распространить на несколько раундов.
Можно считать, что элементы ключа не зависят друг от друга, поэтому преобразование каждого полубайта входа происходит независимо, и можно считать линейные соотношения только для одного полубайта. Для полубайтов K_1, K_2 и H обозначим преобразование $S_{K_1,K_2}(H)$.
$n(K_1, K_2, \alpha, \beta) = (H|0 \leq H < 16, H \times \alpha = S_{K_1,K_2}(H) \times \beta)$
Чем больше количество полубайт H отличается от 8, тем с большей вероятности можно предсказать результат сочетания комбинации (α, β) для пары ключей (K_1, K_2). Например, если $n(K_1, K_2, \alpha, \beta) = 2$, то линейное соотношение выполняется с вероятностью $\frac{2}{16}$, что существенно отличается от $\frac{1}{2}$. Любое отличие более чем на $\frac{1}{4}$ можно считать существенным.

Линейные соотношения для функции перемешивания АШД

Для пары полубайтов ключей (K_1, K_2) назовем удачным линейным соотношением пару (α, β) такую, что $L = |n(K_1, K_2, \alpha, \beta) - 8|$ достигает максимума. Для функции $S_{K_1,K_2}(H)$ были получены следующие результаты:

- Из 192 возможных пар ключей 168 имеют удачные линейные соотношения с $L = 6$, остальные имеют соотношения с $n = 4$

Линейные соотношения для функции перемешивания АШД

Для пары полубайтов ключей (K_1, K_2) назовем удачным линейным соотношением пару (α, β) такую, что $L = |n(K_1, K_2, \alpha, \beta) - 8|$ достигает максимума. Для функции $S_{K_1,K_2}(H)$ были получены следующие результаты:

- Из 192 возможных пар ключей 168 имеют удачные линейные соотношения с $L = 6$, остальные имеют соотношения с $n = 4$
- Из этого можно сделать вывод, что каждое полное преобразование S имеет 32 линейных соотношений, выполняющихся с большой вероятностью, правда сами эти соотношения не известны, если не известны ключи

Линейные соотношения для функции перемешивания АШД

Для пары полубайтов ключей (K_1, K_2) назовем удачным линейным соотношением пару (α, β) такую, что $L = |n(K_1, K_2, \alpha, \beta) - 8|$ достигает максимума. Для функции $S_{K_1,K_2}(H)$ были получены следующие результаты:

- Из 192 возможных пар ключей 168 имеют удачные линейные соотношения с $L = 6$, остальные имеют соотношения с $n = 4$
- Из этого можно сделать вывод, что каждое полное преобразование S имеет 32 линейных соотношений, выполняющихся с большой вероятностью, правда сами эти соотношения не известны, если не известны ключи
- Существует линейное соотношение (5,2), которое в среднем для всех ключей имеет $L = 3$, то есть выполняется или не выполняется с вероятностью $\frac{11}{16}$, что является существенным.

Линейные соотношения для функции перемешивания АШД

Для пары полубайтов ключей (K_1, K_2) назовем удачным линейным соотношением пару (α, β) такую, что $L = |n(K_1, K_2, \alpha, \beta) - 8|$ достигает максимума. Для функции $S_{K_1,K_2}(H)$ были получены следующие результаты:

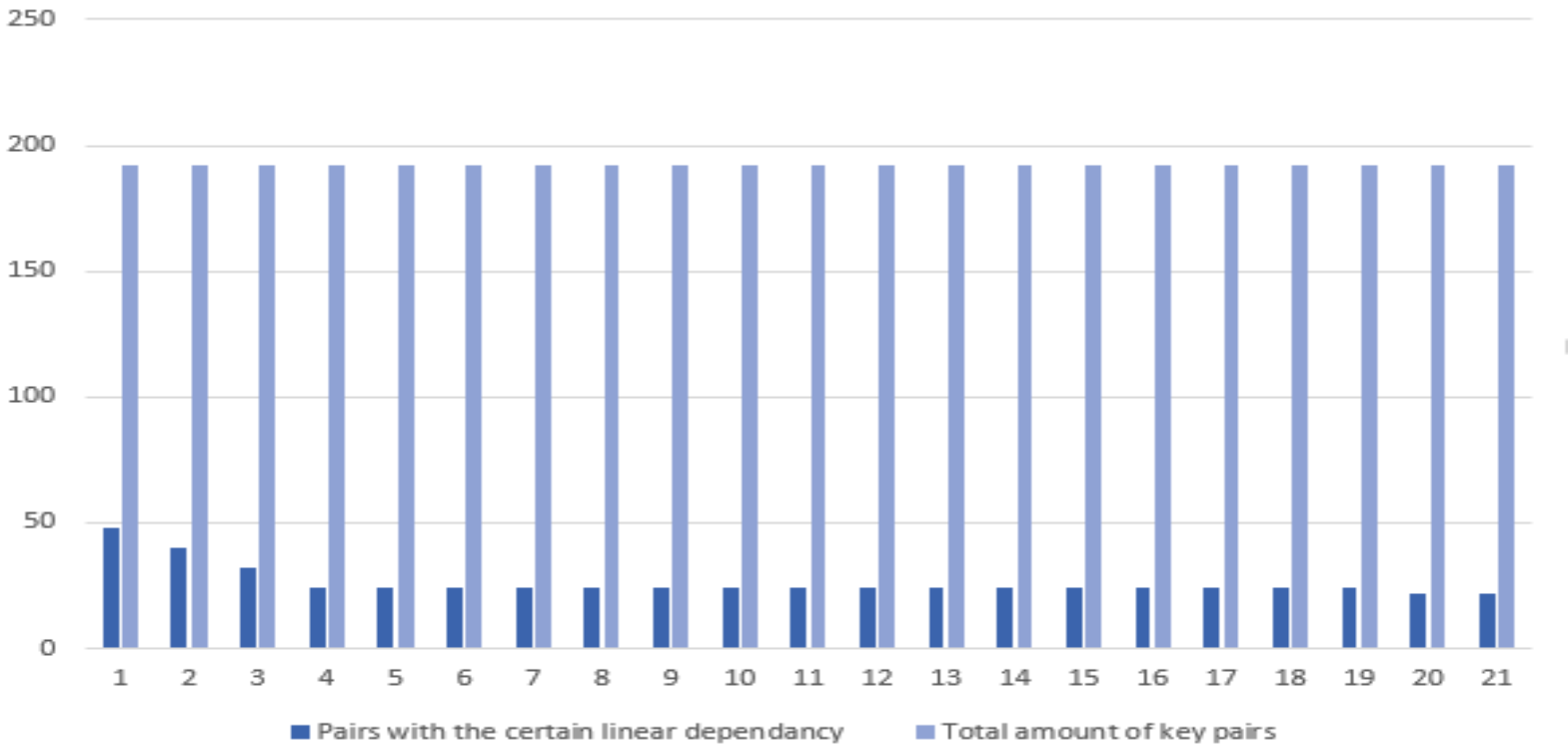
- Из 192 возможных пар ключей 168 имеют удачные линейные соотношения с $L = 6$, остальные имеют соотношения с $n = 4$
- Из этого можно сделать вывод, что каждое полное преобразование S имеет 32 линейных соотношений, выполняющихся с большой вероятностью, правда сами эти соотношения не известны, если не известны ключи
- Существует линейное соотношение (5,2), которое в среднем для всех ключей имеет $L = 3$, то есть выполняется или не выполняется с вероятностью $\frac{11}{16}$, что является существенным.
- Остальные преобразования АШД линейны, то есть соотношения можно продлить на целый раунд, и тогда оно будет включать в себя биты сеансового и раундового ключа

Линейные соотношения для функции перемешивания АШД

Для пары полубайтов ключей (K_1, K_2) назовем удачным линейным соотношением пару (α, β) такую, что $L = |n(K_1, K_2, \alpha, \beta) - 8|$ достигает максимума. Для функции $S_{K_1,K_2}(H)$ были получены следующие результаты:

- Из 192 возможных пар ключей 168 имеют удачные линейные соотношения с $L = 6$, остальные имеют соотношения с $n = 4$
- Из этого можно сделать вывод, что каждое полное преобразование S имеет 32 линейных соотношений, выполняющихся с большой вероятностью, правда сами эти соотношения не известны, если не известны ключи
- Существует линейное соотношение (5,2), которое в среднем для всех ключей имеет $L = 3$, то есть выполняется или не выполняется с вероятностью $\frac{11}{16}$, что является существенным.
- Остальные преобразования АШД линейны, то есть соотношения можно продлить на целый раунд, и тогда оно будет включать в себя биты сеансового и раундового ключа
- распределение удачных соотношений среди пар ключей неоднородны

Распределение удачных соотношений среди пар ключей



Локальные слабости функции перемешивания АШД.

Хотя практическая атака на АШД осуществлена не была, можно сказать,что использование зависящей от ключа функции перемешивания имеет ряд локальных слабостей.

- S-блоки менее устойчивы к линейному криптоанализу, чем специально подобранные.
- Каждая функция $S_{K_1,K_2}(H)$ имеет от 1 до 3х неподвижных точек, то есть теоретически можно подобрать вход алгоритма так, что выход не поменяется.
- Удачный подбор линейных соотношений может дать больше информации о ключе, чем если бы функция перемешивания от ключа не зависела, что становится особенно сильной уязвимостью, если часть информации о ключе или промежуточных этапах шифрования известна.

Некоторых из этих уязвимостей можно было бы избежать, если использовать функцию $S(H[i,j] = (H[i,j]^{-1} \pmod{17}) * K_1[i,j] + K_2[i,j] \pmod{16}$, но это требует дополнительной проверки.

Благодарность

Большое спасибо научным руководителям А.В.Галатенко и А.Е.Панкратьеву за постановку задачи и поддержку в решении.

Литература

[1] Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. Узбекское агентство стандартизации, метрологии и сертификации, Ташкент 2009

[2] Ишматова Ю. А. О некоторых свойствах групп алгебр с параметрами — Интеллектуальные системы, т.15, вып. 1–4, 2011.

[3] Linear Cryptoanalysis method for DES Cipher Mitsuru Matsui, Computer Information Systems Laboratory

[4] Зензин А.С., Иванов М.А. Стандарт криптографической защиты – AES. «Кудиц-образ» Москва 2002

[5] А. А. Карацуба, Сложность вычислений. Труды Математического института им. Стеклова, т. 211, с. 169-183 (1995).

[6] Государственный стандарт Узбекистана - Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ Алгоритм шифрования данных. Узбекское агентство стандартизации, метрологии и сертификации, Ташкент

Конец

Спасибо за внимание!