Умножение с параметром и его применение в криптографии

А. В. Годнева

В работе исследуются свойства операции умножения с параметром, существенно использующейся в криптографических стандартах республики Узбекистан. Изучается структура группы обратимых элементов; устанавливается сложность операции дискретного логарифмирования.

Ключевые слова: мультипликативная группа, криптография, электронная подпись, дискретный логарифм.

1. Введение

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:

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

Эта операция была введена в работах [6, 7]. Несложно увидеть, что для любого R умножение с параметром является коммутативной и ассоциативной операцией, с нейтральным элементом 0.

Пусть $t \in \mathbb{N}$. Обозначим через $a^{\setminus t}$ результат возведения a в степень t относительно введенной операции. В настоящее время возведение в степень относительно умножения с параметром является основой стандарта электронной подписи республики Узбекистан ([2]).

В тексте стандарта [2] без доказательства приводятся критерий обратимости элемента относительно умножения с параметром (элемент a обратим тогда и только тогда, когда $(1 + Ra)$ взаимно просто с n), формула для вычисления обратного элемента и формула для вычисления мощности множества обратимых элементов. Доказательства этих утверждений содержатся в работе [3]. Структура группы обратимых элементов оставалась неизвестной.

В стандарте [2] для вычисления электронной подписи предписывается использовать взаимно простые n и R. В работе [3] показано, что в таком случае возведение с степень с параметром эквивалентно возведению в степень в группе $\mathbb{Z}_n^*$ (под $\mathbb{Z}_n^*$ здесь и далее обозначается группа обратимых по умножению элементов кольца Z_n):

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

Таким образом, с точки зрения стойкости и сложности вычислений алгоритм на основе возведения в степень с параметром оказывается эквивалентным классическим алгоритмам на основе мультипликативных групп вычетов. С другой стороны, несложно увидеть, что в случае, когда $R = 0$, умножение с параметром вырождается в сложение, а дискретное логарифмирование — в деление, и стандарт шифрования на основе такой операции становится уязвимым. Вопрос сложности дискретного логарифмирования для произвольных n и R оставался открытым и полностью решен в этой работе.

Работа имеет следующую структуру. В разделе 2 описывается строение группы обратимых элементов относительно умножения с параметром. В разделе 3 оценивается сложность решения задачи дискретного логарифмирования в случае, когда известно разложение n и R на простые сомножители. В разделе 4 оценивается сложность решения задачи факторизации n и R.

Автор выражает благодарность своему научному руководителю А. В. Галатенко за постановку задачи, помощь и поддержку в процессе решения, а так же А. Е. Панкратьеву за содержательные правки и замечания.

2. Структура группы обратимых элементов

Пусть $n \in \mathbb{N}$ и $0 \leqslant R \leqslant n - 1$. Обозначим через $\mathbb{Z}(n, R)$ группу элементов, обратимых относительно умножения с параметром R. Случай, когда $R = 0$, является тривиальным, поэтому везде далее мы будем считать, что $R \neq 0$. Пусть $a, b \in \mathbb{N}$. Обозначим через $gcd(a, b)$ наибольший общий делитель a и b. Следующая теорема описывает разложение группы обратимых элементов в произведение подгрупп.

Теорема 1. *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* $n_i = p_k^{\alpha_k}$.

Для доказательства нам потребуется ряд вспомогательных утверждений.

Лемма 1. *Пусть* $n = p^\alpha$, $R = p^\beta * r_0$, $gcd(n, r_o) = 1$. *Тогда формула для возведения в степень будет выглядеть следующим образом:*

$$a^{\backslash t} = ta + RC_t^2 a^2 + \ldots + R^{k-1} C_t^k a^k \tag{2}$$

где $k = \lceil \frac{\alpha}{\beta} \rceil$.

Если $\beta \geqslant \alpha$, *то* $k = 1$ *и умножение с параметром в этом случае вырождается в сложение, а возведение в степень — в умножение:* $a^{\backslash t} = ta$.

В случае, когда $\beta = 0$, *операция возведения в степень описывается формулой 1.*

Доказательство. Случаи, когда $\beta = 0$ или $\beta \geqslant \alpha$ либо очевидны, либо уже доказаны, поэтому рассмотрим промежуточный случай. Будем доказывать лемму по индукции.

База $a^{\backslash 1} = a$

Пусть доказываемая формула верна для t. Тогда:

$a^{\backslash t+1} = a + a^{\backslash t} + Raa^{\backslash t} = a + at + RC_t^2 a^2 + \ldots + R^{k-1} C_t^k a^k + Ra^2 t + R^2 C_t^2 a^3 + \ldots + R^k a^{k+1} C_t^k = (t+1)a + R(C_t^2 + C_t^1)a^2 + R^2(C_t^3 + C_t^2)a^3 + \ldots + R^{k-1}(C_t^k + C_t^{k-1})a^k + R^k C_t^k a^{k+1} = (t+1)a + RC_{t+1}^2 + \ldots + R^{k-1} C_{t+1}^k a^k$, где $k = \lceil \frac{\alpha}{\beta} \rceil$.

Слагаемое $R^k C_t^k a^{k+1}$ обращается в 0, так как $R^k = p^{\beta * k} * r_0^k = p^{\beta * \lceil \frac{\alpha}{\beta} \rceil} * r_0^k \equiv \pmod{p^\alpha}$. Таким образом, шаг индукции сделан. Лемма доказана.

Замечание. Рассмотрим общий случай. Пусть $n = \prod\limits_{m=1}^{q} p_m^{\alpha_m} * n_0$, $R = \prod\limits_{m=1}^{q} p_m^{\beta_m} * r_o$, причем простые множители написаны в порядке возрастания и $gcd(n_0, R) = 0$ и $gcd(r_0, n) = 1$. Обозначим $n_i = p_i^{\alpha_i}$ для $i \in \overline{1 \dots q}$, $n = \prod\limits_{i=0}^{q} n_i$. Лемма 1 позволяет решить задачу возведения числа a в степень t отдельно для каждого n_i, то есть получить набор $a_i^{\backslash t}$, такой, что

$$a^{\backslash t} \equiv a_i \pmod{n_i} \forall i.$$

Искомое число $a^{\backslash t}$ можно найти, применяя, например, алгоритм Гарнера [1, Гл. 10, § 10.4] для решения системы модулярных уравнений; единственность и существование этого решения следует из китайской теоремы об остатках и взаимной простоты n_i. Сложность алгоритма Гарнера оценивается сверху числом q^2.

Лемма 2. *Пусть* $gcd(n, R) = 1$. *Тогда* $\mathbb{Z}(n, R) \simeq \mathbb{Z}_n^*$.

Доказательство. Этот факт доказывается непосредственным построением изоморфизма. Элементу $a \in \mathbb{Z}(n, R)$ сопоставим элемент $\overline{a} \in \mathbb{Z}_n^*$ по формуле $\overline{a} = 1 + Ra$. Это отображение является изоморфизмом, так как оно взаимнооднозначно ($a \in \mathbb{Z}(n, R) \Leftrightarrow 1 + Ra \in \mathbb{Z}_n^*$) и сохраняет операцию:

$\overline{a} * \overline{b} = 1 + R(a \circledR b) = \overline{a \circledR b}$.

Лемма доказана.

Лемма 3. *Пусть* $n = p^\alpha$, $R = p^\beta * r$, $0 < \beta < \alpha$, $p \neq 2$. *Тогда группа* $\mathbb{Z}(n, R)$ *циклична, и 1 является ее образующим элементом.*

Доказательство. Все элементы в данной группе являются обратимыми по умножению с параметром, так как для любого a $(1 + Ra) = 1 + p^\beta ra \equiv 1 \pmod{p}$, следовательно $1 + Ra \equiv 1 \pmod{n}$. Таким образом порядок группы будет $|\mathbb{Z}_{(n,R)}| = |Z_n| = p^\alpha$. Тогда 1 будет иметь порядок $T = p^l$, так как порядок элемента делит порядок группы. Воспользовавшись формулой [2], запишем:

$1^{\backslash T} = C_T^1 + RC_T^2 + R^2C_T^3 + \ldots + R^{T-1}C_T^T \pmod n$

Заметим, что это равенство справедливо согласно формуле 2. Если $k < T$, то слагаемые вида $R^{i-1}C_T^i \equiv 0 \pmod n$ для $k < i \leqslant T$. Если $k > T$, то $R^{i-1}C_T^i = 0$ для $T < i \leqslant k$. Таким образом:

$1^{\backslash T} = p^l + C_{p^l}^2 p^\beta r + \ldots + p^{\beta*(p^l-1)} r^{(p^l-1)} \pmod n$

Покажем, что степень p во всех слагаемых, начиная со второго, делятся на p^{l+1}.

Докажем, что $C_{p^l}^s \equiv 0 \pmod{p^{l-\lfloor \log_p s \rfloor}}$

$C_{p^l}^s = C_{p^l-1}^{s-1} * \frac{p^l}{s}$, и среди множителей числа s множитель p встречается не более $\log_p s$ раз, а $C_{p^l-1}^{s-1}$ — целое число, таким образом получили требуемое сравнение. Заметим также, что $\log_p s < s - 1$ при $p \neq 2$, откуда получаем требуемое утверждение:

$1^{\backslash p^l} \neq 0$, если $l < \alpha$. Лемма доказана.

Лемма 4. *Пусть* $n = 2^\alpha$, $R = 2^\beta * r$, $0 < \beta < \alpha$. *Тогда* $\mathbb{Z}(n, R) \simeq \mathbb{Z}_2 \times \mathbb{Z}_{\frac{n}{2}}$ *при* $R = 4s + 2$ *и* $\mathbb{Z}(n, R) \simeq \mathbb{Z}_n$ *при* $R = 4s$.

Доказательство. Аналогично предыдущей лемме, найдем порядок единицы.

$$1^{\backslash 2^l} = C_{2^l}^1 + rC_{2^l}^2 2^{\beta-1} + 2^{l+1} * (\ldots) = 0 \tag{3}$$

При $\beta = 1$ порядок единицы $2^{\alpha-1}$. Докажем, что нет элементов порядка 2^α. Пусть есть и это элемент a. Рассмотрим

$a^{\backslash 2^{\alpha-1}} = a2^{\alpha-1} + a^2 2^{\alpha-1}(2^{\alpha-1} - 1) * r + 2^\alpha(\ldots) = 0$, так как $y + y^2 * x \equiv 0 \pmod 2$ при нечетном x. Получили противоречие с предполагаемым порядком элемента.

Получаем, что при $p = 2$, $R = 4s + 2$, искомая группа имеет вид $\mathbb{Z}(n, R) \simeq \mathbb{Z}_2 \times \mathbb{Z}_{2^{n^{\alpha-1}}}$.

При $\beta > 1$ из формулы 3 видно, что, начиная со второго слагаемого, степени 2 увеличиваются (см. доказательство предыдущей леммы) и порядок единицы равен 2^α, то есть $\mathbb{Z}(n, R)$ также циклична. Лемма доказана.

Перейдем непосредственно к доказательству теоремы 1.

Доказательство теоремы 1. Рассмотрим элемент a, обратимый по умножению с параметром, и сопоставим ему кортеж $(a_0, a_1 \ldots a_q)$, где

$a_0 \equiv a \pmod{n_0}$

$a_1 \equiv a \pmod{n_1 = p_1^{\alpha_1}}$

$\ldots$

$a_q \equiv a \pmod{n_q = p_q^{\alpha_1}}$.

Покажем, что введенное отображение является изоморфизмом $\mathbb{Z}(n, R)$ и прямого произведения групп $\mathbb{Z}(n_i, R)$ по i от 0 до q. Заметим, что все $a_i \in \mathbb{Z}(n_i, R)$. Действительно, это нужно доказывать только для a_0, так как в остальных $\mathbb{Z}(n_i, R)$ обратимы все элементы. Предположим, что a_0 не является обратимым. Тогда $(1 + Ra_0, n_0) \neq 1$. Так как по построению $a = a_0 + n_0 x_0$ для некоторого целого неотрицательного x_0, $1 + Ra = (1 + Ra_0) + Rn_0 x_0$, и следовательно $gcd(1 + Ra, n) \neq 1$, что противоречит предположению об обратимости a.

В силу взаимной простоты n_i отображение является взаимно однозначным (см. [1, Приложение]). Остается заметить, что оно выдерживает умножение с параметром. Возьмем обратимое b и разложим его аналогичным образом. Рассмотрим произведение с параметром a и b.

$a \circledR b = a + b + Rab = a_i + x_i n_i + b_i + y_i n_i + R(a_i + x_i n_i)(b_i + y_i n_i) \equiv a_i \circledR b_i \pmod{n_i}$.

Значит это искомый изоморфизм из $\mathbb{Z}(n, R)$ в произведение $\mathbb{Z}(n_i, R)$, а в каждом $\mathbb{Z}(n_i, R)$ разложение в прямое произведение было уже произведено в леммах 2, 3 и 4.

Итак, нам уже известно, что наша группа обратимых по умножению с параметром элементов определенным образом раскладывается в произведение циклических. Выделим эти циклические подруппы и их образующие непосредственно.

Как и раньше, $n = \prod_{k=1}^{q} p_k^{\alpha_k}$, а $R = \prod_{m=1}^{q} p_m^{\beta_m} * r_0$, и числа n_i такие же, как были рассмотрены ранее. Обозначим через $n'_i = n/n_i = n_0 * \ldots * n_{i-1} * n_{i+1} * \ldots * n_q$. Рассмотрим произвольное n'_i, $i > 0$ $a, b \in 0 \ldots n_i - 1$ Заметим, что:

$(n'_i * a) \circledR (n'_i * b) = n'_i * a + n'_i * b + R * a * b * n'^2_i = n'_i * (a \circledR_1 b) \pmod{n}$,

$R_1 = R * n'_i$ и $a \circledR_1 b$ берется по модулю n_i. Таким образом, от умножения с одним параметром мы перешли к умножению с другим параметром (под $\circledR_1$ обозначается умножение с параметром R_1).

Из предыдущей выкладки можно сделать вывод, что

$\mathbb{Z}(n'_i * a, R) \simeq \mathbb{Z}(n_i, R * n'_i) \simeq \mathbb{Z}_{n_i}$,

где $a \in \mathbb{Z}_{n_i}$

Последний изоморфизм верен при любом $i > 0$, так как соотношение простых делителей при переходе к другому параметру не поменялось.

Таким образом, мы нашли подгруппы в $\mathbb{Z}(n, R)$, изоморфные $\mathbb{Z}_{n_i}$, при любом $0 < i \leqslant q$, а именно это подгруппы, состоящие из элементов $n'_i * a$, $a \in \mathbb{Z}_{n_i}$. Из признака обратимости следует, что все элементы обратимы.

Осталось только рассмотреть модуль n_0. Рассмотрение этого случая аналогично предыдущим, а именно берутся элементы из $\mathbb{Z}_{n_0}$, умножаются на n'_0, из получившегося набора также требуется выделить обратимые по умножению с параметром элементы. В результате, так как в случае, если параметр и модуль взаимнопросты, то умножение с параметром изоморфно обычному умножению в кольце, то получается группа изоморфная $\mathbb{Z}^*_{n_0}$. Таким образом, в нашей группе $\mathbb{Z}(n, R)$ мы явно выделили подгруппы, изоморфные тем, которые нужно, и элементы в этих подгруппах не пересекаются (в пересечении лежит только 0). Значит $\mathbb{Z}(n, R)$ раскладывается в произведение именно этих подгрупп, и каждый элемент может быть представлен в виде произведения их образующих в определенных степенях. Теорема доказана.

3. 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. Имеет место следующее утверждение.

Теорема 2. *Пусть* $n = p^\alpha$, $R = r * p^\beta$, $gcd(r, p) = 1$, $\alpha, \beta \in \mathbb{N}$. *Тогда* $L(n, R) \leqslant c * k^3$, $M(n, R) \leqslant 3k * p^\beta$, *где* $k = \lceil \frac{\alpha}{\beta} \rceil$.

Доказательство. Так как $n = p^\alpha$, $R = rp^\beta$, то можно воспользоваться формулой [2]:

$$a^{\backslash x} = ax + a^2 C_x^2 p^\beta r + a^3 C_x^3 p^{2\beta} r^2 + \ldots + a^k C_x^k p^{\beta(k-1)} \qquad (4)$$

Разложим b, a и x по степеням p^β.

$b = b_0 + b_1 p^\beta + b_2 p^{2\beta} \ldots b_{k-1}^{(k-1)\beta}$

$a = a_0 + a_1 p^\beta + a_2 p^{2\beta} \ldots a_{k-1}^{(k-1)\beta}$

$x = x_0 + x_1 p^\beta + x_2 p^{2\beta} \ldots x_{k-1}^{(k-1)\beta}$

Заметим, что для представления a и b в таком виде достаточно произвести 2k делений с остатком.

Будем последовательно искать $x_0, x_1, \ldots x_{k-1}$, рассматривая $a^{\backslash x}$ по модулям $p^\beta, \ldots, p^{(k-1)\beta}$, p^α соответственно. Алгоритм для нахождения x_l следующий:

Если $gcd(a, n) = p^\gamma$, то и b должно делется на p^γ, иначе рассматриваемое степенное уравнение не имеет решения. Переходим от параметра $R = p^\beta$ к параметру $R_1 = p^{\gamma+\beta}$ и, как несложно убедиться непосредственно, исходное уравнение будет эквивалентно следующему:

$\frac{a}{p^\gamma}^{\backslash x} = \frac{b}{p^\gamma} \pmod n$

Таким образом, можно считать, что $gcd(a_0, p) = 1$.

Если $l = 0$, то $x_0 = \frac{b_0}{a_0}$

Если известны $x_0, \ldots, x_{l-1}$, то

$x_l = \frac{b - a^{\backslash x'} \pmod{p^{(l+1)\beta}}}{a_0}$,

где $x'^l = x_0 + x_1 p^\beta + x_2 p^{2\beta} \ldots x_{l-1}^{(l-1)\beta}$.

Алгоритм останавливается после нахождения всех x_i.

С помощью индукции по i докажем, что в результате выполнения алгоритма вычисляются нужные x_i.

1. база

$a^{\backslash x} = a_0 x_0 \pmod{p^\beta}$

$x_0 = \frac{b_0}{a_0}$

2. Пусть известны $x_0, \ldots, x_{l-1}$. Рассмотрим $a^{\backslash x}$ по модулю $p^{(l+1)\beta}$. С одной стороны, это равно $b \pmod{p^{(l+1)\beta}}$. С другой стороны это значение можно посчитать, используя равенство 4. Чтобы сделать это, нужно вычислить неизвестные $C_x^m \pmod{p^{(l+2-m)\beta}}$, используя определение:

$$C_x^m = \frac{1}{m!}(x_0 + x_1 p^\beta + x_2 p^{2\beta} + \ldots + x_l p^{l\beta})(\ldots)$$
$$(\ldots)(x_0 + x_1 p^\beta + x_2 p^{2\beta} + \ldots + x_l p^{l\beta} - m). \quad (5)$$

Заметим, что для рассмотрения по нужному нам модулю x_l входит только при $m = 1$, так как при $m \geqslant 2$ имеет место:

$$\frac{p^{\beta(m-1)}}{m!} p^{\beta l} \equiv 0 \pmod{p^{(l+1)\beta}} \quad (6)$$

(последнее равенство верно для $p > 2$, для $p = 2$ придется также рассматривать случай $m = 2$)

Говоря строго, из равенства (6) следует, что необходимая нам для подсчета часть равенства 5 выглядит так (при $m > 1$)

$$C_x^m \pmod{p^{(l+2-m)\beta}} = C_{x'^l}^m \pmod{p^{(l+2-m)\beta}}. \quad (7)$$

Из того, что x_l используется только при $m = 1$, следует, что:

$a_0 * x_l = b - a^{\backslash x' l} \pmod{p^{(l+1)\beta}}$, откуда получаем корректность вычисления x_l по предложенному алгоритму.

Индуктивный переход сделан.

Будем проводить верхнюю оценку сложности алгоритма дискретного логарифмирования. Как нетрудно заметить, после декомпозиции n на простые множители, наибольшее количество итераций проводится в случае, если $n = p^k$, $R = p$ (так как при длине входа $n = p^\alpha$, количество x_i, которые нужно найти, равно $\lceil \frac{\alpha}{\beta} \rceil$, то есть максимум достигается при $\beta = 1$)

В этом случае нам необходимо k раз выполнить поиск x_i. Посмотрим, сколько действий нужно выполнить для выполнения i-й итерации.

Согласно предложенному алгоритму для нахождения x_i небходимо вычислить $a^{\backslash x'^i} \pmod{p^{i+1}}$, что выполнимо не более чем за $c_1 * i * \log_2 n \leqslant c_2 * k^2$ операций. Для вычисления $b \pmod{p^{i+1}}$ нужны еще 3 операции.

Для вычисления x достаточно:

$\sum_{i=0}^{k-1} c_2 * k^2 \leqslant ck^3$.

Заметим, что в данном алгоритме достаточно хранить в памяти только элементы a_i, b_i, и x_i, и на них потребуется объем памяти не больше, чем $3 * k * p^\beta$, так как это $3k$ чисел, каждое из которых не больше чем p^β. Теорема доказана.

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

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

то есть задача полиномиальна.

Перейдем к рассмотрению общего случая.

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

Доказательство. Будем, как и ранее, декомпозировать задачу на более простые случаи. Из Теоремы 1 следует, что $a = (n'_0)^{\gamma_0} ® (n'_1)^{\gamma_1} ® \ldots ® (n'_q)^{\gamma_q}$
тогда, так как умножение с параметром коммутативно,
$a^{\backslash x} = (n'_0)^{\gamma_0 x} ® (n'_1)^{\gamma_1 x} ® \ldots ® (n'_q)^{\gamma_q x}$
из вида полученных образующих $n'_0, n'_1, \ldots, n^1_q$ следует, что

$a^{\backslash x} = (n'_0)^{\gamma x} \pmod{n_0}$
$a^{\backslash x} = (n'_1)^{\gamma x} \pmod{n_1}$
.
.
.
$a^{\backslash x} = (n_q)^{\gamma x} \pmod{n_q}$

Таким образом задачу дискретного логарифмирования можно разложить по модулям n_i и решать отдельно для каждого модуля.

Для модуля n_0 задача дискретного логарифмирования сводится к аналогичной для обычного умножения (см. [3]), а для остальных модулей алгоритм дисретного логарифмирования и рассчет его сложности приведены в Теореме 2.

Суммарная сложность алгоритма решения задачи дискретного логарифмирования по каждому из модулей таким образом может быть оценена следующей величиной:

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

А так как сложеность алгоритма Гарнера для решения системы модулярных уравнений оценивается величиной q^2, где q — количество уравнений, то результирующая сложность задачи дискретного логарифмирования:

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

4. Задача факторизации n и R

Для решения задачи дискретного логарифмирования способом, предложенным в третьем разделе, требуется представить n и R в виде произведения простых сомножителей, то есть решить задачу факторизации, имеющей субэкспоненциальную сложность решения (см. [1]). Однако посмотрим, не упростится ли эта задача в рассматриваемом частном 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 4. *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 $M = \prod_{i=1}^{q} p^{gcd(\alpha_i, \beta_i)})$.*

Доказательство. Приведем алгоритм для решения поставленной задачи.

На входе числа n и R, которые нужно представить в следующем виде:

$n = \prod_{k=1}^{q} p_k^{\alpha_k} * n_0$, $R = \prod_{m=1}^{q} p_m^{\beta_m} * r_0$.

Найдем n_0 и r_0 с помощью алгоритма Евклида следующим образом:

$$n^{(1)} = \frac{n}{gcd(n, R)}$$
$$n^{(i)} = \frac{n}{gcd(n^{(i-1)}, R)} \tag{8}$$

Когда процесс стабилизируется, то $gcd(n^t, R) = 1$ получим $n^t = n_0$. r_0 ищется аналогичным образом.

Теперь проведем аналогичную процедуры для чисел, состоящих из одних и тех же простых множителей. Положим $n^{(0)} = \frac{n}{n_0}$, $R^{(0)} = \frac{R}{r_0}$:

Сначала выполним следующую последовательность:

$$n^{(i+1)} = \frac{n^{(i)}}{gcd(n^{(i)}, R)} \tag{9}$$

За шаг до стабилизации (то есть до того, как $gcd(n^{(i)}; R) = n^{(i)}$) фиксируем $n^{(j)}$ и переходим к выполнению следующей последовательности итераций:

$R^{(i+1)} = \frac{R^{(i)}}{gcd(R^{(i)}; n^{(j)})}$

И продолжаем чередование этих процедур до того, как $n^{(l)} = R^{(l)}$. Докажем, что в конце предложенной последовательности операций останется некотрый набор множителей p_i в степенях $gcd(\alpha_i; \beta_i)$.

Будем доказывать это на примере одного простого множителя p. Пусть он входит в $n^{(l)}$ в виде p^γ. Без ограничения общности будем считать, что последней процедурой был «спуск» $n^{(i)}$. Так как $n^{(l)} = R^{(l)}$, то в $R^{(l)}$ p входит в той же степени, и из вида процедуры 9 можно определить, что деление в последнем спуске происходит на p^β (если смотреть только на степень p). Таким образом $n^{(l-1)}$ содержит p в степени $k\beta$. Аналогично доказывается для $R^{(l-1)}$ и так далее. Отсюда видно, что в результате процедуры получается степень p, которая делит и α и β (соответственно степени p в $n^{(0)}$ и $R^{(0)}$), то есть $p^{gcd(\alpha,\beta)}$, так как в степени меньше этой простой делитель входить не может, потому что деление сродни вычитанию степеней.

Отсюда следует, что задача разложения n на множители свелась к задаче разложения некоторого набора множителей из n в степенях $p_i^{gcd(\alpha_i;\beta_i)}$. На процедуры спуска (9) и (8) потребовалось не более

чем $\sum_{i=1}^{q} \alpha_i$ действий, так как каждый шаг сокращает рассматриваемое число как минимум на одно p_i. Отсюда верна оценка сложности, приведенная в теореме 4.

Теперь предположим, что для некоторых простых делителей $gcd(\alpha_i; \beta_i) = \alpha_i$. В этом случае умножение с параметром будет обычным сложением, и если таких простых множителей несколько, то нет необходимости каждый из них выделять отдельно.

Выделить этот модуль (то есть совокупность всех простых множителей, относительно которых n делит R) можно следующим образом:

$$c = \prod p_i^{gcd(\alpha_i;\beta_i)},$$
$$s = \frac{c^2}{gcd(n, c^2)}.$$

Непосредственной проверкой устанавливается, что это и есть необходимый нам отрезок. Теорема доказана.

Замечание. Для того, чтобы осуществить факторизацию в том виде, в котором она нужна для решения задачи дискретного логарифмирования по алгоритму, представленному в Теореме 3, достаточно вместо числа M факторизовать число $M' = \prod_{j=1}^{q'} p_j^{gcd(\alpha_j,\beta_j)}$, где $gcd(\alpha_j, \beta_j) \neq \alpha_i$. Как можно заметить, $M' \leqslant \sqrt{n}$, то есть вычислительная сложность задачи факторизации понижается как минимум в корень раз.

Список литературы

[1] Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. — М.: МЦНМО, 2003.

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

[3] Ишматова Ю. А. О некоторых свойствах групп алгебр с параметрами // Интеллектуальные системы. — 2012. Т. 16, вып. 1–4. — С. 291–298.

[4] Карацуба А. А. Сложность вычислений // Труды Математического института им. Стеклова. — 1995. Т. 211. — С. 169–183.

[5] Коблиц Н. Курс теории чисел и криптографии. — М.: Научное изд-во ТВП, 2001.

[6] Хасанов П. Ф. Модели и алгебры схем цепей и систем / дисс. на соиск. уч. ст. д.т.н. — Ташкент: ТашПИБ, 1975. — С. 144–250.

[7] Хасанов П. Ф. Фигурно-точечные модели и диаопределители матриц. — Ташкент: Укитувчи, 1975.