MATHCAD (МАТКАД)

Маткад есть программное средство, среда (универсальный язык программирования) для выполнения на компьютере математических и технических расчетов.

Маткад располагает большим набором инструментов для работы с числами, формулами, графиками, текстами.

В Маткаде более сотни операторов и логических функций, предназначенных для численного и символьного решения технических проблем различной сложности.

Маткад содержит:

- обширную библиотеку встроенных математических функций;
- инструменты построения графиков различных типов;
- средства создания текстовых комментариев и оформления отчетов.

В Маткаде математические выражения на экране компьютера представляются (пишутся) в общепринятой математической нотации и имеют такой же вид, как в книге, тетради, на доске и т.д.

Основная программная единица в Маткаде есть (Маткад-)функция. (Маткад-)программа есть несколько следующих друг за другом функций, объединенных вертикалью слева.

Функции в программе располагаются и упорядочиваются линейно слева направо сверху вниз.

Всякая функция в программе может вызывать другую функцию. При этом вызываемая функция должна быть расположена не ниже вызывающей функции.

Функция в программе допускает локальные и глобальные параметры. Локальные параметры приоритетны. Если используемые в теле функции параметры не объявлены в имени функции, то эти параметры глобальны и им присваиваются последние вычисленные значения из выше расположенных функций. Если таковых нет, то сообщается, что этот параметр не определен.

Основные операции Маткада, как у всякого языка программирования, есть операции ввода-вывода, операция присваивания и несколько операций циклов. Разница лишь в способах соединения этих операций в одну программу.

Для записи компьютерных программ Маткад имеет много положительного, удобного, практичного, ценимого инженерами.

У Маткада два крупных недостатка.

Первый недостаток присущ подавляющему числу языков программирования, в том числе и Маткаду: отсутствие встроенных алгоритмов для работы с длинными целыми числами. В Маткаде их длина не превосходит пятнадцати десятеричных цифр. Заметим, что язык программирования Питон (Python) не имеет этого недостатка. В Питоне можно работать с целыми числами произвольной длины.

Второй недостаток присущ, по-видимому, одному Маткаду: каждая программа-функция Маткада переносится в текстовые редакторы (например в Word), в виде рисунка, причем функция, как бы длинна она ни была, размещается на одной странице

текстового редактора, причем редактор сильно уменьшает размер шрифта, если длина функции превосходит длину страницы.

Пример. bisection method. Вызвать из Маткада.

Дальнейшее изучение Маткада будем проводить на примерах написания программ из области криптографических алгоритмов защиты компьютерной информации.

Среди криптографических алгоритмов можно выделить два основных. Это шифровальные системы (шифросистемы) и электронные цифровые подписи (ЭЦП). Шифросистемы бывают трех типов. Это блоковые шифры, потоковые (поточные) шифры и шифры с открытым ключом. ЭЦП бывают только с открытым ключом.

В качестве примера мы рассмотрим простую криптосистему с открытым ключом RSA (авторы R.Rivest, A.Shamir , L.Adleman) и реализуем ее на Маткаде. Криптосистема RSA имеет в своем составе три объекта. Это шифросистема, ЭЦП с возвратом сообщения и ЭЦП с хеш-функцией.

Компьютерные системы защиты информации основаны на элементах абстрактной алгебры. Наиболее используемы алгоритмы модулярной арифметики, алгоритмы извлечения дискретного квадратного корня, алгоритмы вычисления дискретного логарифма в группах различного рода.

Криптосистема RSA основывается на модулярной арифметике и на трудности (по времени) факторизации (разложения на простые множители) натуральных чисел.

НЕКОТОРЫЕ АЛГОРИТМЫ МОДУЛЯРНОЙ АРИФМЕТИКИ

Утверждение (деление с остатком). Пусть $b \in \mathbb{N}_+$. Всякое $a \in \mathbb{Z}$. можно единственным образом представить в виде $a = bq + r$, где $q \in \mathbb{Z}, 0 \le r < b$.

Утверждение. Для всяких целых $a \ge 1, h \ge 2$ при некотором $s \ge 0$ существует единственное представление a в виде

$$a = a_s h^s + a_{s-1} h^{s-1} + \cdots + a_1 h + a_0 \tag{1}$$

где $0 \le a_i \le h - 1 (i = 0, 1, \ldots, s-1), 1 \le a_i \le h$

Утверждение. Представление (1) *называется представлением числа a (в системе счисления) по основанию h.* Числа $a_s, a_{s-1}, \ldots, a_0$ называются цифрами числа по основанию h и тогда пишут, что по основанию h число $= (a_s, a_{s-1}, \ldots, a_0)_h$.

Замечание. Представление (1) числа можно рассматривать как полином степени s относительно h, который можно использовать для представления в компьютере больших чисел (порядка нескольких сот цифр в десятеричном представлении) и для производства целочисленных арифметических операций над ними – сложения, умножения, вычитания, нахождения частного и остатка при их делении, переход от одной системы счисления к другой и т.д.

Определение. Натуральное число $p \ge 2$ есть *простое число*, если p делится только на 1 и на p, то есть p не имеет собственных делителей. Целое > 2 есть *составное число*, если имеет собственные делители.

Утверждение. Всякое целое большее единицы число можно *факторизовать* (разложить в произведение положительных простых сомножителей) единственным образом с точностью до порядка сомножителей.

Утверждение. 1. Простые сомножители в факторизации могут повторяться. Пусть $p_1 < p_2 < \ldots < p_k$ есть все различные сомножители в факторизации числа a и a_i есть число вхождений простого p_i в факторизацию. Тогда представление $a = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$ есть *каноническая факторизация* числа a, которая единственна.

3. Иногда каноническая факторизация $a = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$ включает все отсутствующие простые числа между 2 и $_k$ в виде $_0$.

4. Распознавание простоты целого числа с 125 цифрами в его десятеричном представлении существующими методами может быть выполнено в несколько минут. Факторизация такого числа потребует миллионы лет компьютерных вычислений, то есть практически неосуществима.

Определение. Целые числа a и b сравнимы по модулю m (обозначение $a \equiv b (\mathrm{mod}\ m)$), если a и b при делении на m дают один и тот же остаток. Модуль m есть положительное целое число.

Замечание. Свойства сравнений похожи на свойства равенств целых чисел. Есть и существенные различия.

1. Сравнение по одному модулю можно почленно складывать.

1. Сравнение по одному модулю можно почленно складывать.
2. Слагаемые можно переносить из одной части сравнения в другую с обратным знаком.
3. К каждой части сравнения можно прибавить (или вычесть) кратное модулю.
4. Сравнения по одному модулю можно почленно перемножить. Обе части сравнения можно возвести в натуральную степень. Обе части сравнения можно умножить на любое целое.
5. Если
   1) $S = \sum\limits_{a_1,\ldots,a_k} A_{a_1\ldots a_k} x_1^{a_1} \ldots x_k^{a_k}$, где $A_{a_1\ldots a_k}$ есть целые коэффициенты, а сумма $\sum$ берется по конечному множеству наборов $(a_1, \ldots, a_k)$ с натуральными $a_1, \ldots, a_k$.
   2) $A_{a_1\ldots a_k} = B_{a_1\ldots a_k} (\mathrm{mod}\ m)$
   3) $x_i \equiv y_i (\mathrm{mod}\ m), i = 1, 2, \ldots, k,$
   4) $S_1 = \sum\limits_{(a_1,\ldots,a_k)} B_{a_1\ldots a_k} y_1^{a_1} \ldots y_k^{a_k}$, то $S \equiv S_1 (\mathrm{mod}\ m)$.
6. Обе части сравнения $a \equiv b (\mathrm{mod}\ m)$ можно разделить на всякое d если $(d, m) = 1$.
7. Сравнение $a \equiv b (\mathrm{mod}\ m)$влечет $ak \equiv bk (\mathrm{mod}\ mk)$ для всякого целого k, ибо $a \equiv b + mt$, $ak \equiv bk + mkt$.
8. Сравнение $a \equiv b (\mathrm{mod}\ m)$ можно сократить на всякое $d = \mathrm{mod}(a, b, m)$.
9. Если $a \equiv b$ по нескольким модулям, то $a \equiv b$ по их наименьшему общему кратному.
10. Если $a \equiv b (\mathrm{mod}\ m)$, то $a \equiv b (\mathrm{mod}\ d)$для всякого (положительного) делителя d модуля m.
11. . Если одна часть сравнения и модуль делятся на некоторое целое d, то другая часть сравнения делится на d.

12. Если $a \equiv b (\mathrm{mod}\ m)$, то $(a, m) = (b, m)$ (по свойству 11).

Определение. Пусть $a \in \mathbf{Z}_m = \{0, 1, \ldots, m-1\}$. *Мультипликативно обратное* для по модулю m есть целое $a^{-1} \in \mathbf{Z}_m$, для которого $a \cdot a^{-1} (\mathrm{mod}\ m) = 1$. Если такое a^{-1} существует, то оно единственно и a^{-1} называется *обратным* элементом для a.

Вычисление h-ричной записи 10-ричного числа

ВХОД. Натуральные числа $a > 0$ и $h \geq 2$.
ВЫХОД. h-ричная запись числа $a = (a_t a_{t-1} \ldots a_1 a_0)_h$.

1. $i := 0$.
2. Пока $q \neq 0$, выполнять следующее.
   2.1 $r := \mathrm{mod}(a, h), q := (a - r)/h$.
   2.2 $a := q, a_i = r$.
   2.3 $i := i + 1$.
3. Вернуть a.

Вычисление модулярной степени a^k (mod n)

ВХОД. Натуральные числа a, k, n.
ВЫХОД. Степень a^k (mod n).

1. Найти бинарное представление числа $k = k_t k_{t-1} \ldots k_1 k_0$.

1. $b := 1$. Если $k = 0$, то вернуть b.
2. $A := a$.
3. Если $k_0 = 1$, то $b := a$.
4. Для i от 1 до t выполнить следующее:
   (a) $A := A^2 (modn)$
   (b) Если $k_i = 1$, то $b := A * b (modn)$.
5. Вернуть b.

Тест Миллера-Рабина для простоты числа

ВХОД. Нечетное целое $n \geq 3$ и параметр безопасности $t \geq 1$.
ВЫХОД. Ответ "простое"или "составное"на вопрос: "Является ли n простым числом?"

1. Найти s и нечетное r, для которых $n - 1 = 2^s r$.
2. Для i от 1 до t выполнить следующее.
   (a) Выбрать случайное целое a, $2 \leq a \leq n - 1$.
   (b) Вычислить $y := a^r (modn)$.

(c) Если $y \neq 1$ и $y \neq n-1$, то выполнить следующее.
$j := 1$.
Пока $j \leq s-1$ и $y \neq n-1$, выполнить следующее.
Вычислить $y := y^2 (mod n)$.
Если $y = 1$, то вернуть "составное".
$j := j+1$.
Если $y \neq n-1$, то вернуть "составное".

3. Вернуть "простое".

Расширенный алгоритм Евклида вычисления d =нод(a,b) и чисел u, v, для которых $d = au + bv$

ВХОД. Натуральные числа a и b.
ВЫХОД. d =нод(a,b) и целые u, v для которых $d = au + bv$.

1. Если $b = 0$, то $d := a$, $u := 1$, $v := 0$, $return(d, u, v)$.
2. $u_2 := 1$, $u_1 := 0$, $v_2 := 0$, $v_1 := 1$.
3. Пока b > 0 выполнять следующее.
   (a) $q := \lfloor a/b \rfloor, r := a - qb, u := u_2 - qu_1, v := v_2 - qv_1$.
   (b) $a := b, b := r, u_2 := u_1, u_1 := u, v_2 := v_1, v_1 := v$.
4. $d := a, u := u_2, v := v_2$, вернуть (d, u, v)

Вычисление мультипликативного обратного элемента a^{-1} (mod n)

1. С помощью расширенного алгоритма Евклида найти d=нод(a,n) и те целые x и y, для которых $ax + ny = d$.
2. Если d>1, то a^{-1}(mod n) не существует. Иначе обратный элемент $a^{-1} = x$.

КРИПТОСИСТЕМА RSA

Шифросистема RSA

Вычисление ключей. Каждый адресат A вычисляет свой открытый ключ и ему соответствующий секретный ключ. Адресат A должен выполнить следующее.

1. Выбрать два случайных различных простых числа p и q примерно одной длины.
2. Вычислить $n = pq$ и функцию Эйлера $\phi = \phi(n)$=(p-1)(q-1).
3. Выбрать случайное число e, 1<e<ϕ, такое, что НОД(e, ϕ)=1
4. С помощью расширенного алгоритма Евклида вычислить такое целое d, 1<a<ϕ, для которого $ea \equiv 1 (\mathrm{mod} \phi)$.
5. Открытый ключ адресата A есть (n, e). Секретный ключ для A есть a.

Шифрование. Адресат пишет письмо t адресату B и A должен выполнить следующее.

1. Взять открытый ключ (n, e) адресата B.

2. С помощью какого-либо метода M, который публикуется, представить свой текст t как сообщение в виде натурального числа m из сегмента $[0, n\text{-}1]$.

3. Вычислить шифротекст $c \equiv m^e (\mathrm{mod}\ n)$.

4. Отправить шифротекст адресату .

Дешифрование. Чтобы извлечь текст t из 4, B4 должен выполнить следующее.

1. Взять свой секретный ключ a и вычислить $m = c^d (\mathrm{mod}\ n)$.

2. Вычислить текст t адресата с помощью метода M.

Пример шифрования и дешифрования текста

Вычисление ключей. Адресат пишет письмо адресату . Адресат B должен выполнить следующее.

1. Выбрать два случайных различных простых числа примерно одной длины $p = 5903$, $q = 5479$.

2. Вычислить $n = pq = 5903 * 5479 = 32342537$ и функцию Эйлера $\phi = \phi(n) = (p-1)(q-1) = 32331156$.

3. Выбрать случайное число $e = 65537$, $1 < e < \phi$, такое что НОД$(e, \phi) = 1$.

4. С помощью расширенного алгоритма Евклида найти $a = 7832549$ такое, что $ea \equiv 1 (\mathrm{mod}\ \phi)$.

5. Открытый ключ адресата B есть пара чисел ($n = 32342537$, $e = 65537$). Секретный ключ адресата B есть число $a = 7832549$.

Шифрование. Адресат A должен выполнить следующее.

1. Взять открытый ключ ($n = 32342537$, $e = 65537$) адресата B.

2. Представить свой текст $t =$ HOL натуральным числом m с помощью какого-либо метода, например, в 27-ричной системе счисления следующим образом. Пронумеровать буквы алфавита:

Пробел	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	P	Q	R	S	T	U	V	W	X	Y	Z				
	16	17	18	19	20	21	22	23	24	25	26				

Для текста t=HOL вычислить $m = 8 \cdot 27^2 + 15 \cdot 27 + 12 = 6249$.

3. Вычислить $c \equiv m^e (\mathrm{mod}\ \ n) = 6249^{65537} (\mathrm{mod}\ 32342537) = 29967820$

4. Послать шифротекст c адресату B.

Дешифрование. Чтобы дешифровать шифротекст c адресат B должен выполнить следующее.

1. Вычислить (с помощью своего секретного ключа a)
   $m = c^a (\mathrm{mod} n = 301085^{181967} (\mathrm{mod}\ 314869) = 10235)$.

2. Представить число m в 27-ричной системе счисления:
   $m = (81512)_{27}$ и получить исходный текст $t =$ HOL)

Замечание. На практике для криптографической стойкости модуль n задается двоичным числом с 1024 и более бит.

Электронная цифровая подпись (ЭЦП) RSA с помощью хеш-функции

Вычисление ключей. Каждый адресат A создает открытый ключ и ему соответствующий секретный ключ. Адресат A должен выполнить следующее.

1. Выбрать два различных простых числа p и q примерно одной длины.

2. Найти числа $n = pq$ и $\phi = (p-1)(q-1)$.

3. Найти целое число e, $1 < e < \phi$, такое, что НОД$(e, \phi) = 1$.

4. Найте целое число a, $1 < a < \phi$ такое, что $ea \equiv 1 (\mathrm{mod}\ \phi)$

5. Открытый ключ для A есть пара (n, e). Секретный ключ для A есть a.

Вычисление подписи. Адресат A пишет письмо и подписывает свой текст t произвольной длины. Любой адресат B может проверить подпись адресата A. Адресат A должен выполнить следующее.

1. Вычислить значение хеш-функции $h = h(t)$.

2. С помощью своего секретного ключа вычислить подпись $s = h^a (mod n)$ под текстом t.

3. Отправить текст t с подписью s адресату B.

Проверка подписи. Чтобы проверить подпись под письмом t, адресат B должен выполнить следующее.

1. Получить открытый ключ (n, e) адресата A.

2. Вычислить значение хеш-функции $h = h(t)$.

3. Вычислить $h1 = h^s (mod\ n)$. Если нет, то отклонить подпись s.

4. Если $h1 = h$, то принять подпись s. Если $h1 \neq h$, то подпись s отклонить.

Пример ЭЦП RSA с помощью хеш-функции

Вычисление ключей. Адресат A должен выполнить следующее.

1. Выбрать два различных простых числа примерно одной длины $p = 5903$, $q = 5479$.

2. Вычислить $n = p * q = 32342537$ и функцию Эйлера $\phi = (p-1)(q-1) = 32331156$.

3. Выбрать случайное число $e = 65537$, $1 < e < \phi$ такое, что НОД$(e, \phi) = 1$. С помощью расширенного алгоритма Эвклида найти $a = 7832549$ и $x = 12$ такие, что $65537a + x\phi = 1$.

4. Открытый ключ для A есть ($n = 32342537$, $e = 65537$); секретный ключ $a = 7832549$.

Вычисление подписи. Адресат A пишет письмо и подписывает свой текст t. Любой адресат B может проверить подпись адресата A. Адресат A должен выполнить следующее.

1. Вычислить значение хеш-функции $h = h(t)$. Пусть для примера t=HOL, $m = 8 * 27^2 + 15 * 27 + 12 = 6249, h = h(t) = h(m) = m = 6249$.

2. С помощью своего секретного ключа вычислить $s = h^a (mod\ n) = 6249^{7832549} (mod\ 32342537) = 10721215$. Число s есть подпись A под текстом t.

3. Отправить текст t с подписью s адресату B.

Проверка подписи. Чтобы проверить подпись под письмом t=HOL, адресат B должен выполнить следующее.

1. Получить открытый ключ ($n = 32342537$, $e = 65537$) адресата A.

2. Вычислить значение хеш-функции $h = h(t)$. Для этого вычислить $m = 8 * 27^2 + 15 * 27 + 12 = 6249, h = h(t) = h(m) = m = 6249$.

3. Вычислить $h1 = h^s (mod\ n) = 6249^{10721215} (mod\ 32342537) = 6249$.

4. Если $h1 = h$, то принять подпись s. Если $h1 \neq h$, то подпись s отклонить. Так как $h1 = 6249 = h$, то принять подпись s.

Электронная цифровая подпись (ЭЦП) RSA с извлечением сообщения

Вычисление ключей. Каждый адресат A создает открытый ключ и ему соответствующий секретный ключ. Адресат A должен выполнить следующее.

1. Выбрать два различных простых числа p и q примерно одной длины.

2. Найти числа $n = pq$ и $\phi = (p - 1)(q - 1)$.

3. Найти целое число e, $1 < e < \phi$, такое, что НОД$(e, \phi) = 1$.

4. Найти целое a, $1 < a < \phi$ такое, что $ea \equiv 1 (mod\ \phi)$.

5. Открытый ключ для A есть пара (n, e). Секретный ключ для A есть a.

Вычисление подписи. Адресат A пишет письмо адресату B и подписывает свой текст t. Адресат A должен выполнить следующее.

1. С помощью какого-либо метода M, который публикуется, представить свой текст t как сообщение в виде натурального числа m, $0 < m < n - 1$.

2. Найти число $w = R(m)$ с помощью какой-либо открыто публикуемой взаимно однозначной функции R: $[0, n-1] \to M_R$, где M_R есть некоторое числовое множество, например, $R(m) = m * m$, где $a * b$ есть результат прописывания слова b к слову a. Тогда $M_R = \{w = m * m : w \in [0, n-1]\}$.

3. С помощью своего секретного ключа a вычислить $s = w^a (mod\ n)$.

4. Отправить шифротекст s с подписью адресату B.

Проверка подписи и вычисление сообщения. Чтобы проверить подпись и извлечь из нее сообщение адресат B должен выполнить следующее.

1. Получить открытый ключ (n, e) адресата A.

2. Вычислить $w = s^e (mod\ n)$.

3. Проверить, что $w \in M_R$. Если нет, то отклонить подпись s.

4. Найти $m = R^{-1}(w)$.

5. Вычислить текст t адресата A с помощью метода M.

Пример ЭЦП и извлечения текста из подписи

Адресат A пишет письмо t к адресату B и подписывает его. Адресат B может проверить подпись адресата A и извлечь его письмо t.
Вычисление ключей. Адресат A должен выполнить следующее.

1. Выбрать два различных простых числа примерно одной длины $p = 1019, q = 2347$.

2. Вычислить $n = pq = 2391593$ и функцию Эйлера $\phi = (p-1)(q-1) = 1018 * 2346 = 2388228$.

3. Выбрать случайное число $e = 35$, $1 < e < \phi$ такое, что НОД$(e, \phi) = 1$. С помощью расширенного алгоритма Эвклида найти те положительные целые $a = 1569407$ и $x = 12$ такие, что $35a + x\phi = 1$.

4. Открытый ключ для A есть $(n = 2391593, e = 35)$; секретный ключ $a = 1569407$.

Вычисление подписи. Адресат A посылает для B письмо $t = ABX$. Адресат A должен выполнить следующее.

1. Представить текст $t = ABX$ каким-либо способом числом, например, в 27-ричной системе счисления числом $m = 1 * 27^2 + 2 * 27 + 24 = 807$.

2. Вычислить $w = R(m) = R(807) = 807 * 807 = 807807$.

3. Вычислить подпись $s = w^a (mod\ n) = 807807^{1569407} (mod\ 2391593) = 794011$.

4. Отправить шифротекст s адресату B
   Проверка подписи и вычисление сообщения. Адресат B получает шифротекст s и должен выполнить следующее.

   (a) Взять открытый ключ $(n = 2391593, e = 35)$ адресата A.

   (b) Вычислить число $w = s^e (mod n) = 794011^3 5 (mod 2391593) = 807807$.

(c) Так как $w = 807807 = 807 * 807$, то принять подпись и вычислить $m = R^{-1}(w) = 807$.

(d) Представить число $m = 807_{10}$ в 27-ричной системе счисления $m = (1224)_27$ и получает исходный текст $t = ABX$.

ЛИТЕРАТУРА

1. **Болотов А.А., Гашков С.Б., Фролов А.Б. Часовских А.А.** Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы. М.: КомКнига, 2006 – 328 с.

2. **Макаров Е.** . Инженерные расчеты в Mathcad 15. – СПб.: Питер, 2011. 400с.

3. **Очков В.Ф.** Mathcad 14 для студентов и инженеров. – СПб.: БХВ Петербург, 2009. – 512с.

4. **Плис А.И., Сливина Н.А.** Mathcad: математический практикум для экономистов и инженеров. – М.: Финансы и статистика. 1999. 656с.

5. **Набебин А.А.** Дискретная математика. М.: Научный мир, 2010. 512с.

6. **Набебин А.А.** Сборник заданий по дискретной математике. Научный мир, 2009. 280с.

7. **Lidle L., Niederreiter H.** Finite fields. Massach., 1983 – 800p. (Русский перевод: Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир, 1988 – 820p.)

8. **Menezes A., van Oorshot P., Vanstone S.** Handbook of applied cryptography. CRC Press. 1996. 780p.

МАТКАД-ПРОГРАММЫ ДЛЯ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ RSA

Inversion of vector

```
inverse_vect(F) := | n ← last(F)
                   | for i ∈ 0 .. n
                   |   Q_{n−i} ← F_i
                   | Q
```

Representation of 10-based natural numbers as h-based ones

```
h_based_rep(n, h) := | q ← n
                     | i ← 0
                     | while q ≠ 0
                     |     | r ← mod(n, h)
                     |     | q ← (n − r) / h
                     |     | a_i ← r
                     |     | n ← q
                     |     | i ← i + 1
                     | a ← inverse_vect(a)
                     | a
```

Modular exponentiation m^e mod n

```
modexpon(m, e, n) := | c ← 1  if  e = 0
                     | c
                     | break  if  e = 0
                     | c ← m  if  e = 1
                     | c
                     | break  if  e = 1
                     | e2 ← h_based_rep(e, 2)
                     | e1 ← inverse_vect(e2)
                     | t ← last(e1)
                     | c ← 1
                     | A ← m
                     | c ← m  if  e1_0 = 1
                     | for i ∈ 1 .. t
                     |     | A ← mod(A^2, n)
                     |     | c ← mod(A·c, n)  if  e1_i = 1
                     | c
```

Miller-Rabin probalistic primality test

```
MillerRabin(n, t) :=
    if n ≤ 1
        return "Take n>1"
        break
    if n = 2 ∨ n = 3 ∨ n = 5
        return "Prime"
        break
    if mod(n, 2) = 0
        return "Composite"
        break
    n1 ← n − 1
    if mod(n − 1, 2) = 0
        s1 ← 0
        n2 ← 0
        while n2 = 0
            n2 ← mod(n1, 2)
            if n2 = 0
                n1 ← n1 / 2
                s1 ← s1 + 1
        s ← s1
        r ← n1
    for i ∈ 1 .. t
        x ← 1
        while x < 2
            x ← rnd(n − 2)
        a ← round(x)
        y ← modexpon(a, r, n)
        if (y ≠ 1) ∧ (y ≠ n − 1)
            j ← 1
            while (j ≤ s − 1) ∧ (y ≠ n − 1)
                y ← mod(y^2, n)
                if y = 1
                    return "Composite"
                    break
                j ← j + 1
            if y ≠ n − 1
                return "Composite"
                break
        z ← "Prime" if (y = 1) ∨ (y = n − 1)
    return z
```

MillerRabin(48799, 5000) = "Prime"

MillerRabin(131515, 9000) = "Composite"

MillerRabin(32350500, 65537) = "Composite"

MillerRabin(2609, 1000) = "Prime"

Extended Euclidean algorithm to find d=gcd(a,b) and u, v for integers such that d=au+bv

$$\text{gcdxy}(a,b) :=$$

$\text{dxy} \leftarrow \begin{pmatrix} a \\ 1 \\ 0 \end{pmatrix}$ if $b = 0$

dxy

break if $b = 0$

$b = 0$ otherwise

$q \leftarrow 0$

$r \leftarrow 1$

$u \leftarrow 0$

$v \leftarrow 0$

$u2 \leftarrow 1$

$u1 \leftarrow 0$

$v2 \leftarrow 0$

$v1 \leftarrow 1$

$i \leftarrow 1$

while $b > 0$

- $r \leftarrow \text{mod}(a,b)$
- $q \leftarrow \frac{a - r}{b}$
- $u \leftarrow u2 - q \cdot u1$
- $v \leftarrow v2 - q \cdot v1$
- $a \leftarrow b$
- $b \leftarrow r$
- $u2 \leftarrow u1$
- $u1 \leftarrow u$
- $v2 \leftarrow v1$
- $v1 \leftarrow v$
- $i \leftarrow i + 1$
- $\text{dxy} \leftarrow \begin{pmatrix} a \\ u2 \\ v2 \end{pmatrix}$

dxy

$$\text{gcdxy}(3438, 2466) = \begin{pmatrix} 18 \\ 33 \\ -46 \end{pmatrix}$$

$$\text{gcdxy}(4522, 2684) = \begin{pmatrix} 2 \\ -625 \\ 1053 \end{pmatrix}$$

$a := 2^3 \cdot 5 \cdot 7^2$ $\quad b := 2 \cdot 7^2 \cdot 11$

$a = 1960$ $\quad b = 1078$

$$\text{gcdxy}(a,b) = \begin{pmatrix} 98 \\ 5 \\ -9 \end{pmatrix}$$

For integers: The sum, difference, multiplication, division mod p in Zp

$$\text{nmod}(a,p) := \left|\begin{array}{l} c \leftarrow \text{mod}(a,p) \\ c \leftarrow c+p \quad \text{if} \quad c<0 \wedge p \geq 0 \\ c \leftarrow -c-p \quad \text{if} \quad c>0 \wedge p \leq 0 \\ c \leftarrow -c \quad \text{if} \quad c<0 \wedge p \leq 0 \\ c \end{array}\right.$$

$$\text{nmul}(a,b,p) := \left|\begin{array}{l} c \leftarrow a \cdot b \\ c \leftarrow \text{nmod}(c,p) \\ c \end{array}\right.$$

$$\text{nsum}(a,b,p) := \left|\begin{array}{l} c \leftarrow a+b \\ c \leftarrow \text{nmod}(c,p) \\ c \end{array}\right.$$

$$\text{ndif}(a,b,p) := \left|\begin{array}{l} c \leftarrow a-b \\ c \leftarrow \text{nmod}(c,p) \\ c \end{array}\right.$$

Inversion for an integer a and division mod *p* in Z_p

$$\text{rev}(a,p) := \left|\begin{array}{l} \text{if} \quad \text{gcdxy}(a,p)_0 > 1 \\ \quad \left|\begin{array}{l} \text{return} \quad \text{"no inverse"} \\ \text{break} \end{array}\right. \\ c \leftarrow \text{gcdxy}(a,p)_1 \\ \text{while} \quad c<0 \\ \quad c \leftarrow c+p \\ c \leftarrow \text{mod}(c,p) \\ c \end{array}\right.$$

$$\text{rev}(24,31) = 22 \qquad \text{nmul}(24,22,31) = 1$$

$$\text{rev}(9,18) = \text{"no inverse"}$$

$$\text{ndiv}(a,b,p) := \text{mod}(a \cdot \text{rev}(b,p),p)$$

CRYPTOSYSTEM RSA

CIPHER SYSTEM RSA

Key generation

$p := 5903 \quad q := 5479 \quad n := p \cdot q \quad n = 32342537 \quad \phi := (p-1)\cdot(q-1) \quad \phi = 32331156$

$e := 65537 \quad dcx := gcdxy(e, \phi) \quad dcx = \begin{pmatrix} 1 \\ 7832549 \\ -15877 \end{pmatrix}$

$d := \left| \begin{array}{l} d \leftarrow dcx_1 \\ d \leftarrow d + \phi \quad \text{if} \quad dcx_1 < 0 \\ d \end{array} \right. \quad d = 7832549$

Public key for B is the pair of numbers (n, e)
Private key for B is the number d, where $n = 32342537 \quad e = 65537$
$d = 7832549$

Encryption The text t=HOL

$m := 8 \cdot 27^2 + 15 \cdot 27 + 12 \quad m = 6249$ Cypher text $c := modexpon(m, e, n)$

$c = 29967820$

Decryption

$m := modexpon(c, d, n) \quad m = 6249 \quad t1 := h\_based\_rep(m, 27)$

$t1 = \begin{pmatrix} 8 \\ 15 \\ 12 \end{pmatrix}$ The text t = HOL

DIGITAL SIGNATURE RSA with hash-function h(t)

Key generation

$p := 5903 \quad q := 5479 \quad n := p \cdot q \quad n = 32342537 \quad \phi := (p-1)\cdot(q-1) \quad \phi = 32331156$

$e := 65537 \quad dcx := gcdxy(e, \phi) \quad dcx = \begin{pmatrix} 1 \\ 7832549 \\ -15877 \end{pmatrix}$

$a := \left| \begin{array}{l} d \leftarrow dcx_1 \\ d \leftarrow d + \phi \quad \text{if} \quad dcx_1 < 0 \\ d \end{array} \right. \quad a = 7832549$

Public key for A is the pair of numbers (n, e) where $n = 32342537$ $e = 65537$ a = 7832549
Private key for A is the number a,

Signature generation The text t=HOL

$m := 8 \cdot 27^2 + 15 \cdot 27 + 12$ $m = 6249$

$h(m) := m$

$h(m) = 6249$

$s := \text{modexpon}(h(m), a, n)$ $s = 10721215$

Signature verification The text t=HOL

$m := 8 \cdot 27^2 + 15 \cdot 27 + 12$ $m = 6249$

$h(m) := m$

$hm := h(m)$ $hm = 6249$

$s1 := \text{modexpon}(s, e, n)$ $s1 = 6249$

$\text{signature}(s1, hm) :=$ | sgnt ← "signature is accepted" if $s1 = hm$
| sgnt ← "signature is not accepted" if $s1 \neq hm$
| sgnt

$\text{signature}(s1, hm) =$ "signature is accepted"

Remark. If text t is changed, then m1 is other number. For instance,

$m := 3579$ Then

$hm1 := h(m)$ $hm1 = 3579$

$s1 := \text{modexpon}(s, e, n)$ $s1 = 6249$

$\text{signature}(s1, hm1) =$ "signature is not accepted"

DIGITAL SIGNATURE RSA with recovering the message

Key generation

$p := 5903$ $q := 5479$ $n := p \cdot q$ $n = 32342537$ $\phi := (p - 1) \cdot (q - 1)$ $\phi = 32331156$

$e := 65537$ $dcx := gcdxy(e, \phi)$ $dcx = \begin{pmatrix} 1 \\ 7832549 \\ -15877 \end{pmatrix}$

$d :=$ $d \leftarrow dcx_1$ $d \leftarrow d + \phi$ if $dcx_1 < 0$ d $d = 7832549$

Public key for B is the pair of numbers (n, e)
Private key for B is the number d

where $n = 32342537$ $e = 65537$ $d = 7832549$

Signature generation The text t=HO

$m := 8 \cdot 27 + 15$ $m = 231$ $w := 231231$ Signature with text

$s := modexpon(w, d, n)$ $s = 31517476$

Signature verification

$w := modexpon(s, e, n)$ $w = 231231$ Signature is accepted

$m := 231$ $t1 := h\_based\_rep(m, 27)$ $t1 = \begin{pmatrix} 8 \\ 15 \end{pmatrix}$ The text t = HO