АСИММЕТРИЧНЫЕ КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ. СИНТЕЗ, АНАЛИЗ, СВОЙСТВА, ПРИМЕНЕНИЯ

УДК 004 056 55

ОБОСНОВАНИЕ И ИССЛЕДОВАНИЕ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ УЛУЧШЕННОГО АЛГОРИТМА ЦИФРОВОЙ ПОДПИСИ NTRUSIGN

Е.Г. КАЧКО, Д.С. БАЛАГУРА, Ю.И. ГОРБЕНКО

Излагаются результаты обоснования и исследования рациональной практической реализации асимметричного алгоритма цифровой подписи NTRU Sign по критерию минимизации прямых и обратных преобразований.

Ключевые слова: алгоритм NTRU цифровой подписи(Sign), выбор параметров, генерация асимметричной пары ключей, формирование и проверка, сложность алгоритмов формирования и проверки NTRU Sign.

ВВЕДЕНИЕ

В связи с широким применением в настоящее время тщательно исследованы многие аспекты теории и практики цифровых подписей (ЦП) в классе RSA, DSA, ECDSA преобразований, в том числе отечественные стандарты, например, ДСТУ ГОСТ 34 310-2009, ДСТУ 4145-2002[1]. Все эти алгоритмы имеют общий недостаток – они существенно проигрывают в сложности (скорости) преобразований симметричным крипто алгоритмам [1–2]. Кроме этого, при появлении новых математических методов и существенном росте производительности крипто аналитических систем, криптографическая стойкость этих алгоритмов вызывает сомнение. Для повышения стойкости разработчики систем постоянно увеличивают размеры общесистемных параметров для этих алгоритмов. Особенно эта проблема остро встает в связи с разработками, связанными с созданием нейрокомпьютеров, в которых параллельно будут выполняться не только потоки, но и отдельные фрагменты программы (т.н. мелкозернистый параллелизм). Еще большую опасность несет возможное создание квантовых систем крипто анализа, которые будут в состоянии осуществлять атаки типа полное раскрытие для крипто преобразований, сложность которых носит субэкспоненциальный характер. Поэтому важными являются задачи поиска крипто преобразований, которые позволили бы, с одной стороны существенно увеличить скорость криптографических операций, а с другой стороны сохранить криптографическую стойкость. Последние исследования позволяют сделать вывод, что указанным требования могут удовлетворять крыпто преобразования в кольцах срезанных полиномов (NTRU)[2].

Целью статьи является обоснование сущности и разработка рациональной реализации алгоритма NTRU для ЦП, а также сравнения его вычислительной сложности с известными аналогичными асимметричными алгоритмами. При этом, вопросы оптимальности не ставятся, подразумевается некоторое повышение скорости, и такая реализация называется рациональной(улучшенной). Вопросы криптографической стойкости не рассматриваются, а только в отношении сложности криптопреобразований идет ориентация на работы [2-5].

1. ЦИФРОВАЯ ПОДПИСЬ

1.1. Параметры и алгоритмы

Основными этапами реализации ЦП является выбор общих параметров, генерация ключевой пары, а также основные- выработка и проверка ЦП.

Параметры цифровой подписи. Согласно [3] используются следующие параметры ЦП:

N – размер срезанного кольца R, которое определяет максимальную степень полинома (N–1), значения этих параметров составляют 439 и 743 [4];

p, q – малый и больший модули преобразования соответственно, которые являются взаимно простыми числами (для ЦП и для шифрования равны 3, 2048). Кроме того, параметр q определяет интервал, которому должны принадлежать все коэффициенты многочленов, использующихся в криптосистеме. Так, пространство сообщений L_M определяется как:

$$L_M = \{M(x) \in R\},$$

при этом коэффициенты полиномов сообщений лежат в диапазоне

$$[-(q-1)/2,(q-1)/2];$$

дополнительные параметры, которые рассматриваются ниже.

Личный ключ. В отличие от шифрования [2], личный ключ для ЦП содержит 4 полинома:

$f(x), g(x)$ (или f, g) – полиномы по модулю $p = 3$, которые должны иметь обратные элементы. При выборе этих полиномов для обеспечения максимальной безопасности, рекомендуется выбирать только такие, коэффициенты которых имеют заданное число 1 и –1. Эти значения также включаются в список параметров. Обозначим d_f (определяет количество 1, –1 в полиноме f) и d_g (определяет количество 1, –1 в полиноме g).

F, G – полиномы, которые удовлетворяют уравнению:

$$f * G - g * F = q. \quad (1)$$

Необходимо заметить, что значений F, G, удовлетворяющих уравнению (1) множество. Из этого множества следует выбирать наименьшие. Еще одно требование к этим полиномам – нормы векторов для них должны удовлетворять заданному значению, которое обозначается *KeyNormBound* и является параметром. Как следует из [3], всегда можно выбрать эти значения так, чтобы их коэффициенты не превосходили $\sqrt{\frac{N}{12}}$. Для значения *N*, равного 439, максимальное значение коэффициента равно 7, а для *N*, равного 743, максимальное значение коэффициента равно 8. Для минимизации коэффициентов полиномов F, G может использоваться итерационный алгоритм, в котором количество итераций для выбора ограничивается значением параметра *MaxAdjustment*.

Для формирования ЦП достаточно использовать один из полиномов пары (f, g) и один полином из пары (F, G). Выбор элементов пары определяется типом личного ключа. Тип ключа обозначается *basisType* и принимает 2 значения: "*standard*" или "*transpose*". В дальнейшем выбирается тип ключа *transpose*, который позволяет ускорить формирование ключевых данных и ЦП без потери криптографической стойкости[3].

В отличие от алгоритмов ЦП, которые нашли широкое распространение (RSA, DSA, ECDSA, ДСТУ 4145-2004,...), в NTRU открытый ключ используется не только для проверки, но и для ее создания. Поэтому открытый ключ либо вычисляется в процессе выработки ЦП, либо входит в структуру с личным ключом. Последнее более эффективно с точки зрения вычислительной сложности операции создания ЦП. Необходимость использования открытого ключа при создании ЦП связана с тем, что не все ЦП, которые получаются в соответствии с алгоритмом создания, могут быть проверены. ЦП считается «хорошей» не только в случае, если она удовлетворяет определенным математическим соотношениям, но и если норма вектора, соответствующего полиному, полученному в качестве ЦП, удовлетворяет определенным соотношениям. Предельное значение нормы определяется параметром *NormBound*. Поэтому после выработки ЦП она проверяется, и, только в случае успешной проверки, возвращается в качестве значения ЦП. Значение нормы вычисляется по формуле[6]:

$$\|s\|^2 = \sum_{i=0}^{N-1} s_i^2 - \frac{1}{N}\left(\sum_{i=0}^{N-1} s_i\right)^2.$$

При этом, ограничивается количество попыток, которые используются для создания ЦП. Параметр, который определяет число попыток, обозначается *SignFailTolerance*.

Для увеличения криптографической стойкости, создание ЦП выполняется в несколько этапов. Для каждого этапа используется внутренний личный и открытый ключи. Для каждого очередного шага вычисляется новое значение подписываемых данных. Для формирования ЦП на всех этапах необходимо иметь соответствующее количество личных ключей. Поэтому вместо обычной функции генерации личного ключа используется цикл генерации. Для проверки ЦП на стороне получателя используется последний открытый ключ. Количество личных ключей, рекомендуемое для формирования ЦП, задается параметром *perturbationBases*. В работе принимается значение этого параметра равным 1, что соответствует формированию ЦП в 2 этапа:

(for ($i = 0$; $i <= perturbationBases$; $++i$)).

Заметим, что многоуровневое формирование ЦП приводит к увеличению значения нормы на величину $\sqrt{perturbationBases + 1}$.

Открытый ключ. Задается полиномом по модулю *q*. При задании этого полинома используются только положительные числа, отрицательные значения заменяются *q*-значением.

Таким образом, в качестве дополнительных параметров ЦП используются: d_f (задает количество 1, –1 в полиноме f) и d_g (задает количество 1, –1 в полиноме g); *KeyNormBound*, *NormBound* – значения норм векторов, которые соответствуют полиномам (F, G) и ЦП s; *MaxAdjustment* – число итераций для минимизации полиномов F, G; *basisType* – определяет, какие компоненты пар (f, g), (F, G) используются при создании ЦП; *SignFailTolerance* – максимальное число итераций для получения «хорошей» подписи; *perturbationBases* – количество этапов формирования ЦП.

Общие параметры. В табл. 1 заданы значения общих параметров для N = 439 и 743 [4]

Таблица 1

Параметры для полиномов для $N = 439$ и 743

Параметр	N = 439	743
d_f^1 (d_g)	146	248
keyNormBound	280	360
NormBound	400	
MaxAdjustment	439	743
basisType	TRANSPOSE	TRANSPOSE
SignFailTolerance	100	100
perturbationBases	1	1

[1] Значение $d_f(d_g)$ задает количество – 1, количество 1 на 1 больше.

При вычислении ключевой пары необходимо вычислить результанты [5] ключевых полиномов (f,g)) и полинома (X^N-1). Результант – это значение определителя, порядок которого равен 2^N-1 и формируется так (показано для полиномов f и (X^N-1)):

$$\begin{bmatrix} f_{N-1} & f_{N-2} & f_{N-3} & \dots & f_0 & 0 & 0 & 0 & \dots & 0 \\ 0 & f_{N-1} & f_{N-2} & \dots & f_1 & f_0 & 0 & 0 & \dots & 0 \\ \dots & & & & & & & & & \\ 0 & 0 & 0 & \dots & f_{N-1} & f_{N-2} & f_{N-3} & f_{N-4} & \dots & f_0 \\ 1 & 0 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 \\ \dots & & & & & & & & & \end{bmatrix}$$

1.2 Генерация ключей цифровой подписи

Генерация ключей для одного этапа[2]

1. Случайно выбрать полином f, содержащий d_f единиц (1), d_f минус единиц (–1) и остальные 0.

2. Вычислить обратный элемент для $f_{inv}=f^{-1}$, т.е. такой элемент, что $f_{inv}*f=1$ в $(Z/qZ)[X]/(X^N-1)$.

3. Если обратный элемент не существует, то перейти на шаг 1.

4. Вычислить результант $\mathrm{Re}s_f(f,X^N-1)$, а также полином p_f, удовлетворяющий уравнению:

$$p_f*f=\mathrm{Re}s_f(f,Z[X]/X^N-1).$$

5. Случайно выбрать полином g, содержащий d_g единиц (1), d_g минус единиц (–1) и остальные 0.

6. Вычислить обратный элемент для $g_{inv}=g^{-1}$, т.е. такой элемент, что $g_{inv}*g=1$ в $(Z/qZ)[X]/(X^N-1)$. Если обратный элемент не существует, то перейти на шаг 4.

7. Вычислить результант $\mathrm{Re}s_g$ для полинома g и полинома X^N-1, и полином p_g, удовлетворяющий уравнению:

$$p_g*g=\mathrm{Re}s_g(g,Z[X]/X^N-1).$$

8. Решить диафантово уравнение с помощью расширенной теоремы Эвклида:

$$a*\mathrm{Re}s_f+b*\mathrm{Re}s_g=\gcd(\mathrm{Re}s_f,\mathrm{Re}s_g).$$

9. Если $\gcd(\mathrm{Re}s_f,\mathrm{Re}s_g)\neq 1$, то перейти на шаг 1 (наибольший общий делитель, не равный 1, говорит о наличии одинаковых корней у полиномов f, g, что недопустимо).

10. Вычислить полиномы F,G:

$$F=-p_g*B*q \text{ в } (Z/qZ)[X]/(X^N-1)$$

$$G=-p_g*A*q \text{ в } (Z/qZ)[X]/(X^N-1)$$

11. Реверсировать[3] полиномы f,g. Обозначим результат реверсирования f_{rev}, g_{rev} соответственно.

12. Вычислить $t=f*f_{rev}+g*g_{rev}$.

13. Вычислить результант $\mathrm{Re}s_t$ для полинома t и полинома x^N-1, и полином p_t, удовлетворяющий равенству $p_t*t-\mathrm{Re}s_t(Z[X]/(X^N-1))$.

14. Вычислить коэффициенты полнома *c*:

$$c=p_t*(f_{rev}*F+g_{rev}*G) \text{ в } (Z/qZ)[X]/(X^N-1).$$

15. Почленно разделить все элементы полинома *c* на значение $\mathrm{Re}s_t$ с округлением в большую сторону.

16. Вычислить

$$F-=c*f \text{ в } Z[X]/(X^N-1).$$

$$G-=c*g \text{ в } Z[X]/(X^N-1).$$

17. Минимизировать F, G (этот шаг алгоритма необязательный. Требуются дополнительные исследования для проверки целесообразности выполнения этого шага).

17.1 $u=f, v=g$;

17.2 Вычислить

$$E=2N\sum_{i=0}^{N-1}(f^2{}_i+g^2{}_i)-(f(1)+g(1))^2;$$

17.3 for (i=0; i < *MaxAdjustment*;)

a) Вычислить

$$D=4N\sum_{i=0}^{N-1}(F_if_i+G_ig_i)-2(F(1)+G(1))*(f(1)+g(1));$$

b) if (D>E) {F-=u; G-=v; ++i}; else if (D<-E)

{F+=u, G+=v; ++i}

c) u*=X; v*=X в $Z[X]/(X^N-1)$

18. Если

basisType = TRANSPOSE то $f'=F$.

basisType = STANDARD то $f'=F$.

19. Вычислить открытый ключ[4] $h=f\cdot f'$.

Ключи цифровой подписи. После выполнения этого алгоритма *perturbationBases*+1 раз получаем:

Набор личных ключей:

$$f_{perturbationBases}, f'_{perturbationBases}, h_{perturbationBases}$$

...

$$f_0, f'_0, h_0$$

Для значения *perturbationBases*=1 (см. табл. 1) получаем 2 набора ключевых данных:

$$f_1 f'_1, h_1$$

$$f_0 f'_0, h_0$$

Открытый ключ: $h=h_0$

1.3 Формирование цифровой подписи

Алгоритм формирования ЦП состоит из следующих этапов.

1. Преобразование подписываемого сообщения в полином.

[2] Число таких этапов равно *perturbationBases* + 1

[3] Полином называется реверсивным, если коэффициент с индексом 0 не изменяется, а остальные коэффициенты записываются в противоположном порядке, т.е. f1 ↔ fN-1, f2 ↔ fN-2, ...

[4] Полином f' имеет коэффициенты {0,1,-1} для типа TRANSPOSE и целые коэффициенты для типа STANDARD, поэтому генерация открытого ключа и цифровой подписи, которые используют значение f1, выполняется быстрее для типа TRANSPOSE.

2. Вычисление компонентов ЦП.

3. Преобразование компонентов цифровой подписи в строку байтов

1.3.1 *Преобразование подписываемого сообщения в полином*. Преобразование обычно использует значение хеша сообщения, что обеспечивает существенно различные полиномы для близких сообщений.

При реализации для сравнимости результатов используется алгоритм преобразования [4]:

Алгоритм преобразования сообщения в полином.

1. Определить количество байтов для задания одного элемента полинома: $B = \lceil \log_2 q / 8 \rceil$

2. for ($i = 0$; $i < n$; $++i$)

2.1 Вычислить хеш сообщения + i

2.2 Взять в полученном хеше младшие B байт

2.3 Сформировать значение коэффициента полинома равным $\log_2 q \% 8$ бит старшего байта и оставшиеся биты остальных байтов

1.3.2 П*реобразование полинома в массив байтов*

Для каждого коэффициента полинома выполняется его преобразование в битовую строку и конкатенация с предыдущей строкой битов.

1.4 Формирование цифровой подписи

Алгоритм формирования включает в себя формирование цифровой подписи для каждого этапа и суммирование цифровых подписей. Чтобы при проверке цифровой подписи можно было использовать только открытый ключ последнего этапа, на очередном этапе корректируется полином, для которого вычисляется ЦП.

Алгоритм.

Вход: Сообщение, представленное как полином (i)

Параметры: Личный ключ

Выход: При успешном завершении: Успех, Цифровая подпись s – полином, r – число попыток.

В случае ошибки: Ошибка, (за заданное число попыток (*SignFailTolerance*) не удалось сформировать цифровую подпись со значением нормы, не превосходящей *NormBound*).

1) *for (r = 0; r < SignFailTolerance; ++r)*

1.1) $s = 0$

1.2) *for (iLoop = perturbationBases; iLoop>=1; --iLoop)*

1.2.1) $s_{iLoop} = \lceil f_{iLoop} * i / q \rceil - \lceil f'_{iLoop} * i / q \rceil$ [5]

1.2.2) $s += s_{iLoop}$

1.2.3) $i = s_{iLoop} * (h_{iLoop} - h_{iLoop=1})$

1.2.4) $s_{iLoop} = \lceil f_{iLoop} * i / q \rceil - \lceil f'_{iLoop} * i / q \rceil$

1.3) $s += s_{iLoop}$

1.4) *if (норма s < NormBound) break;*

2) *if (r < SignFailTolerance)*

2.1) *Преобразование полинома s в строку байтов sByte*

2.2) *sByte+=байтовое представление (r)*

[5] Из этого шага алгоритма следует, что создание цифровой подписи выполняется быстрее для типа TRANSPOSE, чем для типа STANDARD (см. Предыдущую сноску)

1.5 Проверка цифровой подписи

Алгоритм проверки включает в себя преобразование ЦП в ее компоненты r, s, «расшифрование» ЦП с помощью открытого ключа, и проверку нормы для исходной ЦП и результата

Алгоритм.

Вход: Сообщение, представленное как полином (i)

Параметры: Открытый ключ h; Цифровая подпись (байтовая строка)

Выход: Успех; Ошибка

1) *Распаковка ЦП (s, r)*

2) *Формирование полинома i для заданных значений (s, r)*

3) $t = i - h * s$

4) $\| s \| \le NormBound \;\&\&\; \| t \| \le NormBound$

2. ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ СОЗДАНИЯ И ПРОВЕРКИ ЦИФРОВОЙ ПОДПИСИ

В данном разделе определяется вычислительная сложность функций выработки и проверки ЦП. Полученные результаты сравниваются с аналогичными результатами для формирования и проверки ЦП для RSA и ДСТУ 4145 – 2004 алгоритмов[1,2].

2.1 Выбор параметров цифровой подписи для сравнения.

При выборе параметров авторы исходили из крипто стойкости, обеспечиваемой при применении соответствующих алгоритмов. В [7, 8] определена зависимость уровня безопасности (k) от параметров цифровой подписи для алгоритмов RSA, ECC с простым и двоичными полями и NTRU при p = 3, *perturbationBases*=1 (табл. 2).

Таблица 2

Зависимость параметров алгоритмов от требуемого уровня безопасности

k	80	112	128	192	256
NTRU, N	157	197	223	313	349
RSA, n	1024	2048	3072	7680	15360
ECC, простое поле p	192	224	256	384	521
ECC, двоичное поле m	163	233	283	409	571

Из табл. 2 следует, что полиномы с N>349 для ЦП обеспечивают уровень безопасности больше, чем 256 и по уровню безопасности превосходят все рассмотренные алгоритмы с максимальными значениями параметров, которые используются в настоящее время.

2.2 Сравнение размеров и времен для различных алгоритмов цифровой подписи

Далее приведены результаты сравнения алгоритмов создания и проверки ЦП по двум критериям: размер ЦП (число байт) и время выполнения этих операций (табл. 3). Первая строчка таблицы соответствует результатам, полученным для подписи (N = 439) в Java реализации ([4]), остальные данные получены с помощью

библиотеки авторов (VS 2008, C, Intel R, Pentium (R), Dual CPU E2160 & 1.80 GHz, 0.99 ГБ ОЗУ). Увеличение быстродействия достигнуто за счет максимального использования параллелизма современных процессоров (SSE операции, многоядерность).

Таблица 3

Результаты сравнения размеров и времен для различных алгоритмов ЦП

Алгоритм	Размер цифровой подписи (байт)	Время создания (с)	Время проверки (с)
NTRU ($N = 439$), java [4]	610	2.4	2.4
NTRU ($N = 439$)	610	0,0033	0,0010
RSA[6], $n = 15360$	1920	88,81	0.214
ECC (ДСТУ 4145-2004), двоичное поле $m = 571$	144	0,042	0,169

Таким образом, использование алгоритма NTRU для ЦП позволяет уменьшить время создания и проверки цифровой подписи по сравнению с ДСТУ 4145-2004 более, чем в 10 раз при создания и более, чем в 100 раз при проверке. При этом размер ЦП увеличивается почти в 5 раз. Алгоритм RSA в этом случае проигрывает и по размеру подписи, и по времени ее выработки и проверки.

Для операций выработки и проверки ЦП время существенно различно (разница более, чем в 3 раза). Это связано с особенностями вычислений для обеих операций (см. соответствующие алгоритмы) и необходимостью проверки полученной ЦП (вычисление ее нормы).

Использование для проверки ЦП норм полиномов для ЦП и результата «шифрования» вместо тождеств, как для других алгоритмов ЦП, приводит к тому, что незначительное изменение открытого ключа иногда не приводят к ошибке проверки ЦП. Изменение остальных параметров (сообщения, личного ключа) приводит к ошибке при проверке ЦП.

В тоже время необходимо отметить, что применение ЦП в кольце срезанных полиномов ограничивается проблемой доказательства стойкости. Эта проблема должна рассматриваться отдельно.

Литература

[1] *Горбенко Ю.І., Горбенко І.Д.* Інфраструктури відкритих ключів . Системи ЕЦП. Теорія та практика. Харків. – Форт, 2010. – 593 с.

[2] Polynomial Public Key Establishment Algorithm for the Financial Services Industry.

[3] http://grouper.ieee.org/groups/1363/lattPK/submissions/EESS1v2.pdf

[4] http://grepcode.com/file/repo1.maven.org/maven2/net.sf.ntru/ntru/1.0/net/sf/ntru/sign/SignatureParameters.java

[5] http://ww2.math.uu.se/~svante/papers/sjN5.pdf

[6] http://www.securityinnovation.com/uploads/Crypto/NTRUSign-preV2.pdf

[7] http://www.securityinnovation.com/uploads/Crypto/lll25.pdf

[8] http://books.google.com.ua/books?id=z8nmMkUFqdwC&pg=PA486&lpg=PA486&dq=Security+level+RSA+ECDSA&source=bl&ots=As6_ELXJqP&sig=u67IGY8X4pZtq5kXMLdM8hadn_4&hl=ru#v=onepage&q=Security%20level%20RSA%20ECDSA&f=false

[6] Так как максимальный модуль, для которого определена процедура генерации ключей составляет 4096 (FIPS 186 -3) для определения времени создания цифровой подписи определяется время выполнения операции модульного возведения в степень для заданного значения модуля. При проверке цифровой подписи «открытый ключ»≈216.

Поступила в редколлегию 27.02.2012

Качко Елена Григорьевна, кандидат технических наук, профессор кафедры ПО ЭВМ ХНУРЭ. Область научных интересов: программные средства криптографических систем.

Балагура Дмитрий Сергеевич, кандидат технических наук, доцент кафедры Безопасности информационных технологий ХНУРЭ. Область научных интересов: защита информации, криптографические протоколы выработки и согласования ключей.

Горбенко Юрий Иванович, фото и сведения об авторе см. на с. 187.

УДК 004 056 55

Обґрунтування та дослідження практичної реалізації покращеного алгоритму цифрового підпису NTRUSign / О.Г. Качко, Д.С. Балагура, Ю.І. Горбенко // Прикладна радіоелектроніка: наук.-техн. журнал. – 2012. – Том 11. № 2. – С. 195–199.

Викладаються результати обґрунтування та дослідження раціональної практичної реалізації асиметричного алгоритму цифрового підпису NTRUSign за критерієм мінімізації прямих та зворотних перетворень.

Ключові слова: алгоритм NTRU цифрового підпису (Sign), вибір параметрів, генерація асиметричної пари ключів, формування та перевірка, складність алгоритмів формування та перевірки NTRUSign.

Табл. 03. Бібліогр.: 08 найм.

UDC 004 056 55

Grounding and researching the practical implementation of the improved algorithm of digital signature NTRUSign / E.G. Kachko, D.S. Balagura, Yu.I. Gorbenko // Applied Radio Electronics: Sci. Journ. – 2012. Vol. 11. № 2. – P. 195–199.

Results of grounding and researching the rational practical implementation of the asymmetric algorithm of the digital signature NTRU sign by the criterion of minimizing forward and backword transformations.

Keywords: algorithm of NTRU digital signature (Sign), choice of parameters, generation of an assymetric pair of keys, forming and checking, complexity of algorithms of forming and checking NTRU Sign.

Tab. 03. Ref.: 08 items.