

Информационные технологии и безопасность
АЛГОРИТМЫ РАЗДЕЛЕНИЯ СЕКРЕТА

Інфармацыйныя тэхналогіі і бяспека
АЛГАРЫТМЫ РАЗДЗЯЛЕННЯ САКРЭТУ



УДК

МКС 35.240.40

КП 05

Ключевые слова: секрет, частичный секрет, пороговая схема разделения секрета

Предисловие

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 РАЗРАБОТАН учреждением Белорусского государственного университета «Научно-исследовательский институт прикладных проблем математики и информатики»

ВНЕСЕН Оперативно-аналитическим центром при Президенте Республики Беларусь

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 9 марта 2011 г. № 11 в качестве предварительного государственного стандарта Республики Беларусь со сроком действия с 01.07.2011 по 01.07.2013

3 ВВЕДЕН ВПЕРВЫЕ

4 Срок представления разработчику предстандарта замечаний и предложений, предложений о целесообразности (нецелесообразности) перевода предстандарта в государственный стандарт — до 01.01.2013

Адрес: 220030, г. Минск, пр. Независимости, 4

Факс: (017) 2095104

Телефон: (017) 2095071

E-mail: apmi@bsu.by

Содержание

1	Область применения	1
2	Нормативные ссылки	1
3	Термины и определения	1
4	Обозначения	2
5	Соглашения	2
5.1	Слова	2
5.2	Слова как многочлены	3
6	Общие положения	3
6.1	Назначение	3
6.2	Пороговая схема разделения секрета	4
6.3	Секрет, промежуточный секрет и частичные секреты пользователей	4
6.4	Открытые ключи	5
7	Алгоритмы пороговой схемы разделения секрета	5
7.1	Алгоритмы генерации параметров	5
7.2	Алгоритм разделения секрета	6
7.3	Алгоритм восстановления секрета	6
Приложение А (справочное) Пример реализации (3,5)-пороговой схемы разделения секрета		8
Приложение Б (рекомендуемое) Модуль АСН.1		10
Приложение В (справочное) Таблицы открытых ключей		11
Приложение Г (справочное) Вспомогательные алгоритмы		14
Библиография		15

**ПРЕДВАРИТЕЛЬНЫЙ ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ**

**Информационные технологии и безопасность
АЛГОРИТМЫ РАЗДЕЛЕНИЯ СЕКРЕТА**

**Інфармацыйныя тэхналогіі і бяспека
АЛГАРЫТМЫ РАЗДЗЯЛЕННЯ САКРЭТУ**

Information technology and security
Secret sharing algorithms

Дата введения 2012-07-01

Дата окончания действия 2013-07-01

1 Область применения

Настоящий предстандарт определяет семейство криптографических алгоритмов разделения секрета (ключа) между пользователями таким образом, что лишь заранее определенные подмножества пользователей могут восстановить его.

Настоящий предстандарт применяется при разработке средств криптографической защиты информации.

2 Нормативные ссылки

В настоящем предстандарте использована ссылка на следующий технический нормативный правовой акт в области технического нормирования и стандартизации (далее – ТНПА):

ГОСТ 34.973-91 (ИСО 8824-87) Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1)

Примечание — При пользовании настоящим предстандартом целесообразно проверить действие ТНПА по каталогу, составленному по состоянию на 1 января текущего года, и по соответствующим информационным указателям, опубликованным в текущем году.

Если ссылочные ТНПА заменены (изменены), то при пользовании настоящим предстандартом следует руководствоваться замененными (измененными) ТНПА. Если ссылочные ТНПА отменены без замены, то положение, в котором дана ссылка на них, применяется в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем предстандарте применяют следующие термины с соответствующими определениями:

3.1 пороговая схема разделения секрета; пороговая СРС: Совокупность алгоритмов разделения и восстановления секрета, реализующих (k, t) -пороговую структуру доступа.

3.2 (k, t) -пороговая структура доступа: Все подмножества множества из t пользователей, которые включают не менее k пользователей.

3.3 пороговое число: Число пользователей k , достаточное для восстановления секрета.

3.4 секрет: Конфиденциальные данные, которые разделяются между пользователями.

3.5 промежуточный секрет: Конфиденциальные данные, получаемые в ходе выполнения алгоритмов разделения/восстановления секрета.

3.6 частичный секрет пользователя: Конфиденциальные данные пользователя, используемые для восстановления секрета.

3.7 открытый ключ: Открытые данные, используемые для восстановления секрета.

3.8 октет: Двоичное слово длины 8.

3.9 неприводимый многочлен: Многочлен, не являющийся постоянным, который нельзя представить в виде произведения многочленов меньшей степени.

4 Обозначения

В настоящем предстандарте применяют следующие обозначения:

$\{0, 1\}^n$	множество всех слов длины n в алфавите $\{0, 1\}$;
$\{0, 1\}^{n*}$	множество всех слов в алфавите $\{0, 1\}$, длина которых кратна n ;
$u \parallel v$	конкатенация $u_1u_2 \dots u_nv_1v_2 \dots v_m$ слов $u = u_1u_2 \dots u_n$ и $v = v_1v_2 \dots v_m$;
$01234 \dots_{16}$	представление $u \in \{0, 1\}^{4*}$ шестнадцатеричным словом, при котором последовательным четырем символам u соответствует один шестнадцатеричный символ (например, $10100010 = A2_{16}$);
\mathbb{F}_2	поле из двух элементов 0 и 1;
$\mathbb{F}_2[x]$	кольцо многочленов над полем \mathbb{F}_2 ;
$u(x)$	а) для $u = u_1u_2 \dots u_8 \in \{0, 1\}^8$ многочлен $u_1x^7 + u_2x^6 + \dots + u_8$ и б) для $u = u_1 \parallel u_2 \parallel \dots \parallel u_n$, $u_i \in \{0, 1\}^8$, многочлен $u_1(x) + x^8u_2(x) + \dots + x^{8(n-1)}u_n(x)$;
$\deg f(x)$	степень многочлена $f(x) \in \mathbb{F}_2[x]$;
$f(x) \bmod m(x)$	для многочлена $f(x) \in \mathbb{F}_2[x]$ и ненулевого $m(x) \in \mathbb{F}_2[x]$ остаток от деления $f(x)$ на $m(x)$;
$f(x) \operatorname{div} m(x)$	для многочлена $f(x) \in \mathbb{F}_2[x]$ и ненулевого $m(x) \in \mathbb{F}_2[x]$ частное от деления $f(x)$ на $m(x)$;
$\text{НОД}(f(x), g(x))$	наибольший общий делитель многочленов $f(x)$ и $g(x)$, т. е. такой многочлен $d(x)$, что $f(x) = d(x)q_1(x)$ и $g(x) = d(x)q_2(x)$, и все общие делители $f(x)$ и $g(x)$ делят $d(x)$;
$a \leftarrow u$	присвоение переменной a значения u .

5 Соглашения

5.1 Слова

Входными и выходными данными алгоритмов, описанных в настоящем предстандарте, являются двоичные слова. Двоичные слова представляют собой последовательности символов из алфавита $\{0, 1\}$. Символы нумеруются слева направо. Например, в слове

$$w = 10110001100101001011101011001000$$

первый символ — 1, второй — 0, последний — 0.

Слова разбиваются на тетрады последовательных двоичных символов. Тетрады кодируются шестнадцатеричными символами по следующим правилам (см. таблицу 1):

Таблица 1

тетрада	символ	тетрада	символ	тетрада	символ	тетрада	символ
0000	0 ₁₆	0001	1 ₁₆	0010	2 ₁₆	0011	3 ₁₆
0100	4 ₁₆	0101	5 ₁₆	0110	6 ₁₆	0111	7 ₁₆
1000	8 ₁₆	1001	9 ₁₆	1010	A ₁₆	1011	B ₁₆
1100	C ₁₆	1101	D ₁₆	1110	E ₁₆	1111	F ₁₆

Например, слово w кодируется следующим образом:

$$B194BAC8_{16}.$$

Пары последовательных тетрад образуют октеты. Последовательные октеты слова w имеют следующий вид:

$$10110001 = B1_{16}, 10010100 = 9A_{16}, 10111010 = BA_{16}, 11001000 = C8_{16}.$$

5.2 Слова как многочлены

Оклету $u = u_1u_2 \dots u_8$ ставится в соответствие многочлен $u(x) = u_1x^7 + u_2x^6 + \dots + u_8$. Многочлен ставится в соответствие также любому непустому двоичному слову из целого числа октетов. Используется соглашение «от младших к старшим»: первому оклету соответствует многочлен $u_1(x)$, второму — $x^8u_2(x)$, третьему — $x^{16}u_3(x)$ и т. д.

Для многочлена $f(x) \in \mathbb{F}_2[x]$ и ненулевого многочлена $m(x) \in \mathbb{F}_2[x]$ определено деление (x) на $m(x)$, которое заключается в представлении $f(x)$ в следующем виде:

$$f(x) = q(x)m(x) + r(x), \quad \deg r(x) < \deg m(x).$$

Многочлен $r(x)$ является остатком от деления, а многочлен $q(x)$ — частным от деления.

6 Общие положения

6.1 Назначение

Настоящий предстандарт определяет семейство криптографических алгоритмов, предназначенных для распределения секрета между t пользователями. Для некоторого фиксированного числа $k \leq t$ все подмножества пользователей, состоящие не менее чем из k пользователей, образуют так называемые разрешенные подмножества. Они и только они могут восстановить секрет. Остальные подмножества пользователей называются запрещенными и не получают никакой дополнительной информации о секрете, кроме априорной.

Пороговая схема разделения секрета, определенная в настоящем предстандарте, содержит следующие алгоритмы:

- а) алгоритмы генерации параметров;
- б) алгоритм разделения секрета;
- в) алгоритм восстановления секрета.

Алгоритмы генерации параметров предназначены для генерации открытых ключей по заданным параметрам t и длине секрета n в октетах. Данные алгоритмы используют случайные или псевдослучайные числа.

Алгоритм разделения секрета предназначен для генерации частичных секретов пользователей для заданных секрета, порогового числа k и открытых ключей. Данный алгоритм использует случайные или псевдослучайные числа. Для их выработки должен использоваться физический генератор случайных чисел, удовлетворяющий ТНПА, или алгоритм генерации псевдослучайных чисел с секретным параметром, определенный в ТНПА.

Алгоритм восстановления секрета позволяет разрешенным подмножествам пользователей получить исходный секрет по их частичным секретам и открытым ключам. Данный алгоритм является детерминированным.

В приложении А приведен пример реализации (3,5)-пороговой схемы разделения секрета. Данный пример можно использовать для проверки корректности реализации алгоритмов настоящего стандарта.

В приложении Б приводится модуль абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ 34.973. Модуль задает идентификаторы алгоритмов настоящего стандарта. Рекомендуется использовать модуль при встраивании алгоритмов, определенных в настоящем пред-стандарте, в информационные системы, в которых также используется АСН.1.

6.2 Пороговая схема разделения секрета

В настоящем предстандарте определена схема разделения секрета, реализующая (k,t) -пороговую структуру доступа. При этом:

- каждый пользователь $i \in \{2, \dots, t\}$ получает в личное пользование и хранение частичный секрет S_i , которому соответствует открытый ключ M_i . Кроме того, всем пользователям доступен общий открытый ключ M_0 ;
- разрешенные подмножества пользователей восстанавливают истинное значение секрета S , если открытые ключи и частичные секреты этих пользователей корректны;
- пороговая схема разделения секрета предусматривает возможность использования одних и тех же открытых ключей для различных секретов, равных по длине, и различных k .

Данная пороговая схема разделения секрета обладает также следующими свойствами:

- для запрещенных подмножеств пользователей вероятность того, что алгоритм восстановления секрета вернет истинное значение секрета длины N бит равна 2^{-N} . Это свойство называется совершенностью;
- длина частичного секрета пользователя совпадает с длиной секрета. Это свойство называется идеальностью.

Общее число пользователей t и длина секрета N в битах должны удовлетворять неравенству $tN \leq 2^{N-1}$.

6.3 Секрет, промежуточный секрет и частичные секреты пользователей

Секрет является двоичным словом определенной длины N , кратной 8.

При выполнении алгоритмов разделения/восстановления секрета строится промежуточный секрет с использованием случайного слова q . По завершении выполнения алгоритмов промежуточный секрет и слово q должны быть уничтожены.

Частичные секреты пользователей являются двоичными словами той же длины, что и секрет. Они должны храниться и распространяться с соблюдением мер конфиденциальности и контроля целостности. Утрата частичных секретов в количестве, меньшем порогового числа k , не влияет на безопасность исходного секрета и не влечет необходимость его замены.

6.4 Открытые ключи

Открытые ключи M_0, M_1, \dots, M_t являются двоичными словами, длина которых совпадает с длиной секрета N . При распространении и хранении открытых ключей должен обеспечиваться контроль их целостности.

Открытым ключам соответствуют многочлены $f_0(x) = x^N + M_0(x), f_1(x) = x^N + M_1(x), \dots, f_t(x) = x^N + M_t(x)$. Многочлены $f_i(x)$ должны быть попарно взаимно простыми. Данное условие будет автоматически выполнено, если они различны и неприводимы. В 7.1 определено два алгоритма генерации открытых ключей: алгоритм генерации открытых ключей, которым соответствуют неприводимые многочлены (7.1.3), и алгоритм генерации открытых ключей, которым соответствуют произвольные попарно взаимно простые многочлены (7.1.4). Первый алгоритм предпочтительнее использовать при значениях $t < 20$ и $N \leq 256$, где $N = 8n$.

Некоторые возможные значения открытых ключей приведены в приложении В.

7 Алгоритмы пороговой схемы разделения секрета

7.1 Алгоритмы генерации параметров

7.1.1 Входные и выходные данные

Входными данными алгоритмов генерации параметров являются:

- число пользователей t ;
- длина секрета n в октетах.

Выходными данными алгоритмов генерации параметров являются слова $M_0, M_1, \dots, M_t \in \{0, 1\}^{8n}$. Слово M_i является открытым ключом пользователя $i \in \{1, \dots, t\}$, а слово M_0 — общим открытым ключом.

7.1.2 Переменные и вспомогательные алгоритмы

Алгоритм тестирования на неприводимость `IsIrred` берет на вход многочлен $f(x) \in \mathbb{F}_2[x]$ и возвращает 1, если $f(x)$ неприводим, и 0 в противном случае. Данный алгоритм должен быть детерминированным. Варианты алгоритма представлены в [1], [2], а также в приложении Г.

7.1.3 Алгоритм генерации открытых ключей по неприводимым многочленам

Для генерации открытых ключей выполняются следующие шаги:

- 1 Установить $i \leftarrow 0$.
- 2 Пока $i < t + 1$, выполнить:

1) Выработать с помощью генератора случайных или псевдослучайных чисел слово $M_i \in \{0, 1\}^{8n}$.

2) Если $\text{IsIrred}(x^{8n} + M_i(x)) = 1$, то:

а) Для всех $j = 0, \dots, i - 1$ проверить, что $M_j \neq M_i$.

б) Если совпадений не найдено, то $i \leftarrow i + 1$.

3 Возвратить (M_0, M_1, \dots, M_t) .

7.1.4 Алгоритм генерации открытых ключей по попарно взаимно простым многочленам

Для генерации открытых ключей выполняются следующие шаги:

1 Установить $i \leftarrow 0$.

2 Пока $i < t + 1$, выполнить:

1) Выработать с помощью генератора случайных или псевдослучайных чисел слово $M_i \in \{0, 1\}^{8n}$.

2) Для всех $0 \leq j < i$ проверить, что $\text{НОД}(x^{8n} + M_j(x), x^{8n} + M_i(x)) = 1$.

3) Если условие предыдущего шага выполняется, то $i \leftarrow i + 1$.

3 Возвратить (M_0, M_1, \dots, M_t) .

7.2 Алгоритм разделения секрета

7.2.1 Входные и выходные данные

Входными данными алгоритма разделения секрета являются:

- число пользователей t ;
- пороговое число k ;
- длина секрета n в октетах;
- секрет $S \in \{0, 1\}^{8n}$;
- открытые ключи $M_0, M_1, \dots, M_t \in \{0, 1\}^{8n}$.

Выходными данными алгоритма разделения секрета являются частичные секреты пользователей $S_1, S_2, \dots, S_t \in \{0, 1\}^{8n}$.

7.2.2 Переменные

Переменными являются слово $q \in \{0, 1\}^{8(k-1)n}$, и промежуточный секрет $C \in \{0, 1\}^{8kn}$. Они должны быть уничтожены сразу после использования.

7.2.3 Алгоритм разделения секрета

Для разделения секрета S на частичные секреты S_1, S_2, \dots, S_t выполняются следующие шаги:

1 Выработать с помощью генератора случайных или псевдослучайных чисел слово q .

2 $C(x) \leftarrow (x^{8n} + M_0(x))q(x) + S(x)$.

3 Для $i = 1, 2, \dots, t$ выполнить $S_i(x) \leftarrow C(x) \bmod (x^{8n} + M_i(x))$.

4 Возвратить (S_1, S_2, \dots, S_t) .

7.3 Алгоритм восстановления секрета

7.3.1 Входные и выходные данные

Входными данными алгоритма восстановления секрета являются:

- число пользователей l в подмножестве $A = \{i_1, i_2, \dots, i_l\}$;

- длина секрета n в октетах;
- открытые ключи $M_0, M_{i_1}, \dots, M_{i_l} \in \{0, 1\}^{8n}$;
- частичные секреты пользователей $S_{i_1}, S_{i_2}, \dots, S_{i_l} \in \{0, 1\}^{8n}$.

Выходными данными алгоритма восстановления секрета является секрет $S \in \{0, 1\}^{8n}$ или сообщение «ОШИБКА».

Для успешного восстановления секрета должно выполняться условие $l \geq k$.

7.3.2 Переменные и вспомогательные алгоритмы

Переменными являются промежуточный секрет $C \in \{0, 1\}^{8ln}$, слова $g \in \{0, 1\}^{8ln}$ и $u, v, d \in \{0, 1\}^{8(l-1)n}$. Промежуточный секрет должен быть уничтожен сразу после использования.

Расширенный алгоритм Евклида Eu берет на вход многочлены $f(x), g(x) \in \mathbb{F}_2[x]$ и возвращает многочлены $d(x), u(x), v(x) \in \mathbb{F}_2[x]$, такие что $d(x) = \text{НОД}(f(x), g(x)) = u(x)f(x) + v(x)g(x)$. Варианты алгоритма представлены в [1], а также в приложении Г.

7.3.3 Алгоритм восстановления секрета

Для восстановления секрета подмножеством пользователей $A = \{i_1, i_2, \dots, i_l\}$ выполняются следующие шаги:

- 1 Установить $j \leftarrow 1$, $C(x) \leftarrow S_{i_1}(x)$, $g(x) \leftarrow x^{8n} + M_{i_1}(x)$.
- 2 Пока $j < l$, выполнить:
 - 1) $j \leftarrow j + 1$.
 - 2) $(d(x), u(x), v(x)) \leftarrow \text{Eu}(g(x), x^{8n} + M_{i_j}(x))$.
 - 3) Если $d(x) \neq 1$, то вернуть сообщение «ОШИБКА».
 - 4) $C(x) \leftarrow (u(x)g(x)S_{i_j(x)+v(x)}(x^{8n} + M_{i_j}(x))C(x)) \bmod (g(x)(x^{8n} + M_{i_j}(x)))$.
 - 5) $g(x) \leftarrow g(x)(x^{8n} + M_{i_j}(x))$.
- 3 $S(x) \leftarrow C(x) \bmod (x^{8n} + M_0(x))$.
- 4 Возвратить S .

Приложение А

(справочное)

Пример реализации (3,5)-пороговой схемы разделения секрета

В настоящем приложении рассматривается пример реализации (3,5)-пороговой схемы разделения секрета.

Для секрета

$$S = 5F891BE8340B60FC95E70A930635B525F8A5C610A7A7CE9582BCDA6A12C86047_{16}$$

длиной 32 октета и открытых ключей M_i , приведенных в таблице А.1, сгенерировано слово q (таблица А.2) и получен промежуточный секрет C (таблица А.3). В таблице А.4 содержатся частичные секреты пользователей S_i , а в таблице А.5 представлены результаты выполнения алгоритма восстановления секрета для различных подмножеств пользователей.

Таблица А.1 — Открытые ключи

Номер ключа	Открытый ключ M_i
0	794BC27C9324A1A51ECE5CB7CA36C0963C55E0DB46D9CC0AEF123FFC53811243 ₁₆
1	9101C43E7D348974CA2C34F2F273854F10CF0C75767AD05EDFF279A40F45690C ₁₆
2	076673ACBEC88478216BCE590E9B3E79B23F14B3B0BCB22E9CB0881469B6CC7 ₁₆
3	DD78F5F22D62B128FD5E180A4272AC842A9B671040A6E915CFC2EE9A12FAFD73 ₁₆
4	6FDBEC49C48E09E8B6D7C9402EF73AB43BA1239D8436FE4EAF36D4F355766BC7 ₁₆
5	537071CA876B7D568E77562D73FCCAC32CC24C84F0D806A43AC0A63A7D2193E5 ₁₆

Таблица А.2

Слово q
6746D27E011F379E96EBDE722FB79BF291D635606E04E93379A77732C65DFF7F CC860839FCE207AACC6704520D1BD5EBA557DD23E71A4278B6F8224143D165AD ₁₆

Таблица А.3

Промежуточный секрет C
509E4844CE8E6C2E8FE1143C50DFD03D8BAD89D5DAC3EA6A847C633657B5BE3F CA34C1CC472BB56494B222932E88D5052A9A54A1D51A18CFE34F864243D92480 17C728366FB785862EF71370C63819B413CBD36D6DA824C619FFBA72C9D1C087 ₁₆

Таблица А.4

Номер пользователя	Частичный секрет S_i
1	864C32754331B8680DF497AFB26A5DA26C694D7C898331165097B3A25126BE64 ₁₆
2	EDD67862260CEC457B33D9AEBE3A82134584A03C441794C36623DA00EC4285A2 ₁₆
3	06DB03FCB3E7AAE0EDECEC8BABAA675B2DAC6FAFB36918827EFA1566A63F18E5 ₁₆
4	F8AEB99BDAFDD1E7A98AC9F1BAE2EB989F94105DD23B016397EE821F5C952D77 ₁₆
5	96D3E320E17317936D841CC9212C465E1E5DB7EA40B53549B987A2D21400E98D ₁₆

Таблица А.5 — Результаты работы алгоритма восстановления секрета

Подмножество пользователей	Восстановленное значение
{1, 2}	070B8B9DA544CB6293AD7A8B9DC97D2E58ADFCE77A3D7B15BD34CD4906A67C5E ₁₆
{1, 3}	F74820D1CB998B089F2A52376D243B5D545270F74DA55D4F741881D25AC16950 ₁₆
{1, 4}	58D2731A10C01A9D4CBE22154FB4C19B69D17933254CD01F195CE444008B2F3B ₁₆
{1, 5}	9AC1D79E469427FEE1C3ED20A787088853A9F4D1AE5333B43EEFDC44AA68010A ₁₆
{2, 3}	F0864B61A46F2753344822950BF0B17A31A90CD1B9FE01D095049A9C4F711E60 ₁₆
{2, 4}	E5E710A6EA5C136E08B4FDE21D1F5E3440C9DF272A383B6AD799BCFAB791488A ₁₆
{2, 5}	989169ECB5D1E021169824EF12A0FC1969B47FOEC02EFAAF4C51220F033A3E87 ₁₆
{3, 4}	DE06BF2287BE547661572FB20CA901A6D20E3BD8D1B7650F263FB2E03D9560B6 ₁₆
{3, 5}	40F61CA6124453CCCDF59C36F71D7CD11EA98191365B3A6D4943054FD47E4E55 ₁₆
{4, 5}	6C68F274A07A1EAB7AE081454A3F58DB49C40CBF7E31ED82802626C157A4616F ₁₆
3 и более пользователей	5F891BE8340B60FC95E70A930635B525F8A5C610A7A7CE9582BCDA6A12C86047 ₁₆

Приложение Б
(рекомендуемое)
Модуль АСН.1

Алгоритмы, определенные в настоящем предстандарте, обозначаются следующим образом:

- | | |
|------------------------|---|
| bels-genirred | алгоритм генерации открытых ключей по неприводимым многочленам (7.1.3) |
| bels-gencoprime | алгоритм генерации открытых ключей по попарно взаимно простым многочленам (7.1.4) |
| bels-share | алгоритмы разделения и восстановления секрета (7.2.3, 7.3.3) |

Модуль АСН.1 имеет следующий вид:

```
Bels-module-v1 {iso(1) member-body(2) by(112) 0 2 0 34 101 44 module(1) ver1(1)}
DEFINITIONS ::=
BEGIN
  bels OBJECT~IDENTIFIER ::= {iso(1) member-body(2) by(112) 0 2 0 34 101 44}

  bels-share OBJECT IDENTIFIER ::= {bels 11}
  bels-genirred OBJECT IDENTIFIER ::= {bels 101}
  bels-gencoprime OBJECT IDENTIFIER ::= {bels 102}
END
```

Приложение В

(справочное)

Таблицы открытых ключей

Таблицы В.1 – В.3 содержат шестнадцатиричное представление открытых ключей $M(x)$, соответствующих неприводимым многочленам $f(x) = x^N + M(x)$, $\deg M(x) < N$. В таблице В.1 представлено 30 открытых ключей для $N = 128$, в таблице В.2 – для $N = 192$, а в таблице В.3 – для $N = 256$.

Таблица В.1 — Открытые ключи $M(x)$, соответствующие неприводимым многочленам $f(x) = x^{128} + M(x)$

Номер ключа	Открытый ключ $M(x)$
1	15221157A5FA7D4FC1D2A1F269E23497 ₁₆
2	2140AC7D5CF07194947DEE9B90E2F0DD ₁₆
3	F1040B9311830F014FA296F3AFDAD5E9 ₁₆
4	A5A8EB19E7CDEAD8E14DFF3858E9CB2E ₁₆
5	F3F09606F1FA0B46B68B27CF35B10157 ₁₆
6	2927986651941EFA9BEBE4E2A3B88761 ₁₆
7	87DBFBEDAA05878253CEB2AC397C59FE ₁₆
8	6B198B7A253176D61275E7C17CF1AF5D ₁₆
9	5153403AF06CFAEDAABE11F0612E5777 ₁₆
10	03F71881C77B7D8F7B8781FFCA058C35 ₁₆
11	57D2032B705603A2A387B753F716C2F0 ₁₆
12	8FF2ACB442B14DA1781A70603FDCD60D ₁₆
13	2764A0F5B2B2EBD2136E7B31CA17D64A ₁₆
14	8B12787E3EB2FE6A1DC1C35655308851 ₁₆
15	3962156CC33868DF984170D2A8ED5DC2 ₁₆
16	8B22FF66817FBA1C956BEF1CF9E21C9D ₁₆
17	33E6E6D9B41B4B9AEC6B9789BC72D973 ₁₆
18	BD45860BCE6C2C860A7F85D72E7AC640 ₁₆
19	D58CC2DF5B60E0F6A52692204EB56C3C ₁₆
20	63DB5F2A67DF54FB78867C6F662BBB43 ₁₆
21	95FD6D128F47683B595FA1024D7F4484 ₁₆
22	BDB4691F706F939946ADFFBA78AE0715 ₁₆
23	5D4F6A191DC0575461C4540A150C33FD ₁₆
24	6BF6AA8129F91398D07F608A501A8278 ₁₆
25	7D7FBCA5C954FC7D39CAFE9787277DFE ₁₆
26	7BB010941820399016B4AED58BB731B2 ₁₆
27	B17F515CB45DB4A435DDA17475334657 ₁₆
28	BBBC7AC705631D4FE7F26427076D5458 ₁₆
29	A31E266089B08652C0D221346F60B80F ₁₆
30	790B759BA7B498B8DFDA9D8A43268C80 ₁₆

Таблица В.2 — Открытые ключи $M(x)$, соответствующие неприводимым многочленам $f(x) = x^{192} + M(x)$

Номер ключа	Открытый ключ $M(x)$
1	29B2D4AEE0785CA6FD0DCCE20FA2F43F3784F29158D2E6D8 ₁₆
2	A1F89FA48286CA6C23C8CF90C63F7B9560B89DBCA2DDE000 ₁₆
3	AF3C9932B4D5A4F79F5B33D6F54ACD47BED2B1327FFB4577 ₁₆
4	47569E452170475F01E960BD0D96F64590CBB08758896302 ₁₆
5	A1A275C35152AB676D4E9D68ADD0091605E118440CDF7907 ₁₆
6	170AF79BD7637B34623C1DB3B0161090069EF4D358F359F5 ₁₆
7	851405F39C1647CC9709A1366B4B6D7EA3D7B843EFD95DDD ₁₆
8	2D26F89829D6D63A1D996EC55423F9114233F4544F39CFC3 ₁₆
9	09D250694D2E1FE54B71A76952EEC9A217C2F28D4B1B952A ₁₆
10	B1F1B8CFD514032ED8DE269DF56EEC2B291303F48421BF28 ₁₆
11	35B8FBE24D6C965768DD7243CD56B701A4AD3EA77B651646 ₁₆
12	E543C2BC3EE387D9B00AFB38A66A987BB040418AA3C53028 ₁₆
13	B7A4D88314410F9C8F028990D58C649C427EB8B957980705 ₁₆
14	85E40EEABCAC0789B70A5C33E769431FD892ADD526A7DBB9 ₁₆
15	27BE59EB589FAFD62FDC5480DBCAECBC7152BAF0B06DB772 ₁₆
16	D18ED523D40A5DA75B61682FB46533C04758F38DDE3280C7 ₁₆
17	6B3B481374A72748AFC68C10641E91A76F18DA5E55935444 ₁₆
18	CF7BEB5644F83C3838E58D2B6AC75B37E48B4AC82F52014B ₁₆
19	0BFF044C35AC68A9EEE79A2A3DB8C4A93B75DBB8823BB727 ₁₆
20	6BB10FCCE5BAE593C721524F9ECB1517D1F096B5719A1D1F ₁₆
21	41F2EC7487535B8B8FBAEC8D87B7F0B8CDD86638BB9F9630 ₁₆
22	03028D0A3ABE7D6A970B4522417FDA169E367D385E81FC88 ₁₆
23	1DFE66452DF48619431CE21E38EB7A39C5D9D4A687354D82 ₁₆
24	FF5CA6522ECEDCC8195D1DDF3C0872B019EDB72AF07C8531 ₁₆
25	EF765CB19F5D051BFA286EEA39A4A17D6F501E770A34338E ₁₆
26	41FE23A3B3243AF62D5AFE556C89BAABDE6D493EE87B1144 ₁₆
27	BF40C261BD12CE2C3F0C7D63B5142FCD43ED976F96FC7286 ₁₆
28	A1340CBCF627F61E95AFB3EBF0F9D9382850BF3160B9E280 ₁₆
29	91C41056F70B617794D0B9BC1FE98707282C85613CB6B767 ₁₆
30	DDA0BBC4C0E9FFAB1059D83020E7AC5F0282E451DE5A4C11 ₁₆

Таблица В.3 — Открытые ключи $M(x)$, соответствующие неприводимым многочленам $f(x) = x^{256} + M(x)$

Номер ключа	Открытый ключ $M(x)$
1	794BC27C9324A1A51ECE5CB7CA36C0963C55E0DB46D9CC0AEF123FFC53811243 ₁₆
2	9101C43E7D348974CA2C34F2F273854F10CF0C75767AD05EDFF279A40F45690C ₁₆
3	076673ACBEC88478216BCE590E9B3E79B23F14B3B0BCB22E9CB0881469B6CC7 ₁₆
4	DD78F5F22D62B128FD5E180A4272AC842A9B671040A6E915CFC2EE9A12FAFD73 ₁₆
5	6FDBEC49C48E09E8B6D7C9402EF73AB43BA1239D8436FE4EAF36D4F355766BC7 ₁₆
6	537071CA876B7D568E77562D73FCCAC32CC24C84F0D806A43AC0A63A7D2193E5 ₁₆
7	4723A12E23B0E0734C99FF3014423543B7CC1C64018D093F3D9CC9CDBA387AC3 ₁₆
8	0DF8EDB3AFD058B00E48DFOF8625ABAE08700EBDD158CDCD6A62DE5C21957A42 ₁₆
9	498030A57B1E2D3A821A01B75C927A6226483E8E380149B8BE1FA4079BE6EFEB ₁₆
10	ADD8A52C94FB3970342ECF6616C2CD9A2F387A90A012A34F876E9F7CDF89F1C6 ₁₆
11	4B7AAA4F4B1E2C05B2C1151B85B3281967391E9F51DFC6D223D1EB4329211ED8 ₁₆
12	71FAF09548D1D1CD36996C121169439113C18549D27DFEAFD4DD3174D68B97F ₁₆
13	A38434946B3EE8AAD757244B3C69D30A29A6208F8CD94E1A33C087693D7AD93D ₁₆
14	655B2D259875D8AD8E17FF51DBD5C401FCA6C4D301AF24FF486B69336B693AA6 ₁₆
15	FFE21B93229C9876BC98E203A0F84F759C53A0EB50030C867FADB1F83D1797BF ₁₆
16	EBAF498033F00A16EBCFE9228FF25029C4C80025CC68197A6F784B2772463D62 ₁₆
17	D3F041769B45CBF90B8EA9213851036260E010C98C25DCDE215C3A6637268616 ₁₆
18	0F22DA80C00901B817E8BE6491C48B1647DD9413E31FCEBA2CF9BE176D877FDB ₁₆
19	F1B0080A60D97B26D3514F22BD70E9BB6B2ED24B33689E087EE99735580D8EF5 ₁₆
20	0B575D32AB5F7CB7F13D6DB2C8996494BBB27979442295098F4980C74341C567 ₁₆
21	2D8D637CEAAE40669A96C1AD77E7344CE51807CADF80BEE83A1D2A17CCE9B023 ₁₆
22	5B9654C1533B2F6D05ACEBA3111F184DF9B92800A1CC003D1776AA5D4FFA4D7C ₁₆
23	E916F1767D063E679DB47550D03B6E4976931202DEF4E1D0D177D78BE398F3E2 ₁₆
24	1DEFE19D9A0F83FC985045E6EBDE913DFE549C5B49C875A67D2DF9DB447A087E ₁₆
25	C1BA18BA7B80CC3DA48B05D2142D65984C7F91F7E1FE32E1530E76F5AC9ACBA0 ₁₆
26	2DEDFOF94D1457AA47C176E6AB22D8ADB656B259731F68BC5FD3BEC1B32955BF ₁₆
27	BB8F931B5460B1B8E233421ED0683B35E0B47B44A54B6951A320981ADDE6F6AB ₁₆
28	819508C608007EDFA31DCAADD369D1A86A3649CBOCA25E9455565E8BB2021ECE ₁₆
29	597718691E34694D3CABAB8470997E76F270B11BC93D0D52AE698A34E31E4367 ₁₆
30	E923C67A5D15C6D02BAD6A3EB4A1BE987BE8714E1ACCA0F0D0F5005CC79CFF08 ₁₆

Приложение Г

(справочное)

Вспомогательные алгоритмы

Настоящее приложение содержит примеры вспомогательных алгоритмов IsIrred (7.1.2) и Eu (7.3.2).

Г.1 Алгоритм тестирования на неприводимость IsIrred

Г.1.1 Входные и выходные данные

Данный алгоритм берет на вход многочлен $f(x) \in \mathbb{F}_2[x]$ и возвращает 1, если $f(x)$ неприводим, и 0 в противном случае.

Г.1.2 Переменная

Используется переменная p со значениями в $\mathbb{F}_2[x]$, $\deg p(x) < \deg f(x)$.

Г.1.3 Алгоритм

Для тестирования многочлена $f(x) \in \mathbb{F}_2[x]$ на неприводимость выполняются следующие шаги:

- 1 Установить $p(x) \leftarrow x^2$, $i \leftarrow 1$.
- 2 Пока $i < \deg f(x)$, выполнить:
 - 1) Если i делит $\deg f(x)$ и $\text{НОД}(p(x) + x, f(x)) \neq 1$, то вернуть 0.
 - 2) $p(x) \leftarrow p^2(x) \bmod f(x)$.
 - 3) $i \leftarrow i + 1$.
- 3 Если $p(x) \neq x$ вернуть 0, в противном случае вернуть 1.

Г.2 Расширенный алгоритм Евклида Eu

Г.2.1 Входные и выходные данные

Алгоритм берет на вход многочлены $f(x), g(x) \in \mathbb{F}_2[x]$, $\deg f(x) \geq \deg g(x)$, и возвращает многочлены $d(x), u(x), v(x) \in \mathbb{F}_2[x]$, такие что $d(x) = \text{НОД}(f(x), g(x)) = u(x)f(x) + v(x)g(x)$.

Г.2.2 Переменные

Используются переменные $t_1, t_2, t_3, u, v, d, u_1, v_1, d_1, q$ со значениями в $\mathbb{F}_2[x]$.

Г.2.3 Алгоритм

Для нахождения многочленов $d(x), u(x)$ и $v(x)$ выполняются следующие шаги:

- 1 Установить $(t_1(x), t_2(x), t_3(x)) \leftarrow (0, 0, 0)$.
- 2 Установить $(u(x), v(x), d(x)) \leftarrow (1, 0, f(x))$.
- 3 Установить $(u_1(x), v_1(x), d_1(x)) \leftarrow (0, 1, g(x))$.
- 4 Пока $d_1(x) \neq 0$, выполнить:
 - 1) $q(x) \leftarrow d(x) \text{ div } d_1(x)$;
 - 2) $(t_1(x), t_2(x), t_3(x)) \leftarrow (u_1(x), v_1(x), d_1(x))$;
 - 3) $(u_1(x), v_1(x), d_1(x)) \leftarrow (u(x) + q(x)u_1(x), v(x) + q(x)v_1(x), d(x) + q(x)d_1(x))$;
 - 4) $(u(x), v(x), d(x)) \leftarrow (t_1(x), t_2(x), t_3(x))$.
- 5 Возвратить (d, u, v) .

Библиография

- [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

- [2] Лидл Р., Нидеррайтер Г. Конечные поля
М.: Мир, 1988