МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Методические указания к выполнению лабораторных работ по дисциплине

«Криптографические методы защиты информации»
для студентов специальности:
6-05-0611-03 «Искусственный интеллект»

Брест 2023

УДК 347 77/681.3
ББК 67.403.3 73/32.97

В методических указаниях приведены необходимые теоретические сведения по основам применения алгоритмов хеширования в криптографии. Методические указания содержат информацию о протоколах обмена ключевой информацией с использованием эллиптических кривых. Содержатся алгоритмы и указания для выполнения протоколов слепой и коллективной электронной цифровой подписи. Также содержится информация об основных протоколах тайного голосования.

Методические указания предназначены для использования студентами специальности 6-05-0611-03 «Искусственный интеллект» в ходе выполнения лабораторных работ по дисциплине «Криптографические методы защиты информации».

Составители: Хацкевич М. В., старший преподаватель кафедры ИИТ
Глущенко Т. А., старший преподаватель кафедры ИИТ
Соловчук А. М., старший преподаватель кафедры ИИТ
Хацкевич А. С., преподаватель - стажер кафедры ИИТ

Рецензент: Грицук Д. В., заведующий кафедрой прикладной математики и информатики учреждения образования «Брестский государственный университет» им. А. С. Пушкина, кандидат физико-математических наук, доцент

Учреждение образования

СОДЕРЖАНИЕ

1 ХЕШ-ФУНКЦИЯ

ОСНОВНЫЕ СВЕДЕНИЯ

Функция хеширования H(m) или хеш-функция (hash-function) – это детерминированная функция, на вход которой подается строка битов произвольной длины, а выходом всегда является битовая строка фиксированной длины n.

Значение хеш-функции H(m) для входа m называют хешем.

Исходная строка m, для которой вычислено хеш-значение, называется прообразом хеш-функции.

1. Стойкость к поиску первого прообраза – отсутствие эффективного полиномиального алгоритма вычисления обратной функции, т. е. нельзя восстановить текст m по известной его свертке H(m) за реальное время (необратимость). Это свойство эквивалентно тому, что хеш-функция является односторонней функцией.

2. Стойкость к поиску второго прообраза (коллизиям первого рода) – вычислительно невозможно, зная сообщение m и его свертку H(m), найти такое другое сообщение $m' \neq m$, чтобы $H(m') = H(m)$.

3. Стойкость к коллизиям (коллизиям второго рода) – коллизией для хэш-функции называется такая пара значений m и m' , $m' \neq m$, для которой $H(m') = H(m)$.

Стойкость хеш-функции к коллизиям означает, что нет эффективного полиномиального алгоритма, позволяющего находить коллизии.

Замечание: свойства не являются независимыми:

– Обратимая функция неустойчива к восстановлению второго прообраза и коллизиям.

– Функция, нестойкая к восстановлению второго прообраза, нестойка к коллизиям; обратное неверно.

– Функция устойчивая к коллизиям, устойчива к нахождению второго прообраза.

– Устойчивая к коллизиям хеш-функция не обязательно является односторонней.

Важно, чтобы значения хеш-функции сильно изменялись при малейшем изменении аргумента, т. е. ей должен быть присущ лавинный эффект.

Значение хеша не должно давать утечки информации даже об отдельных битах аргумента.

ПРИМЕНЕНИЕ ХЕШ-ФУНКЦИЙ ДЛЯ ПРОВЕРКИ ИСТИННОСТИ СООБЩЕНИЙ

Основные шаги процесса хеширования:

– Отправитель A подает сообщение на вход функции хеширования и находит его свертку (хеш).

– Свертка сообщения добавляется к сообщению.

– Отправитель A отправляет получателю сообщение+свертку .

– Получатель пропускает сообщение через функцию хеширования.

– Получатель B генерирует свое собственное значение свертки сообщения и сравнивает две свертки сообщения. Если они совпадают, сообщение не было изменено.

Для более высокого уровня защиты надо использовать код аутентификации сообщений (MAC – Message Authentication Code).

ТИПЫ КРИПТОГРАФИЧЕСКИХ ХЕШ-ФУНКЦИЙ

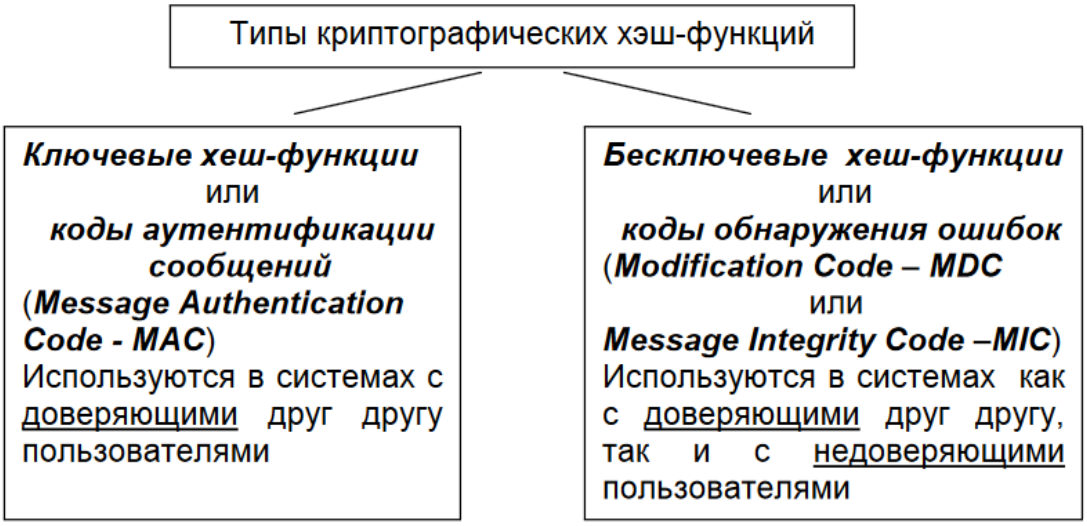


Рисунок 1.1 – Типы криптографических функций

Разработка хеш-функций, удовлетворяющих всем требованиям, – сложная задача. Практически отвечают этим требованиям хеш-функции из группы алгоритмов SHA.

Алгоритм безопасного хеширования (SHA – Secure Hash Algorithm) – стандарт, разработанный NIST. SHA основан на схеме Меркеля-Дамгарда.

Алгоритм	Описание	Длина хеш,бит
SHA-1, SHA-256, SHA-384, SHA-512	Односторонняя функция	SHA-1 создает хеш длиной 160 бит, SHA-256 – длиной 256 бит и т. д

АЛГОРИТМ БЕЗОПАСНОГО ХЕШИРОВАНИЯ SHA-1

Основные характеристики алгоритма:

– Длина хеш-кода: 160 бит.
– Длина обрабатываемых блоков: 512 бит.
– Число шагов алгоритма: 80 (4 раунда по 20 шагов).
– Максимальная длина хешируемых данных: (2^{64} –1).

Алгоритм получает на входе сообщение максимальной длины (2^{64}-1) бит и создает в качестве выхода дайджест сообщения длиной 160 бит.

Алгоритм состоит из следующих шагов:

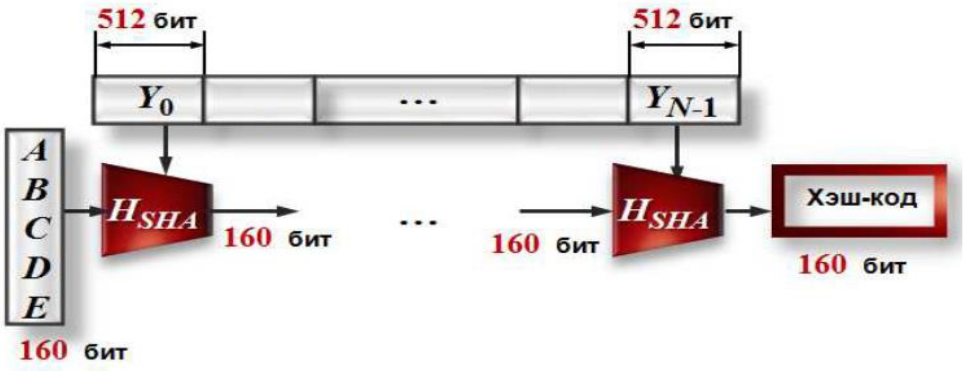


Рисунок 1.2 – Логика SHA-1

Исходное сообщение разбивается на блоки по 512 бит в каждом. Последний блок дополняется до длины, кратной 512 бит. Сначала добавляется 1 (бит), а потом – нули, чтобы длина блока стала равной 512 – 64 = 448 бит. В оставшиеся 64 бита записывается длина исходного сообщения в битах (в big-endian формате). Если последний блок имеет длину более 447, но менее 512 бит, то дополнение выполняется следующим образом: сначала добавляется 1 (бит), затем – нули вплоть до конца 512-битного блока; после этого создается ещё один 512-битный блок, который заполняется вплоть до 448 бит нулями, после чего в оставшиеся 64 бита записывается длина исходного сообщения в битах (в big-endian формате). Дополнение последнего блока осуществляется всегда, даже если сообщение уже имеет нужную длину.

Инициализируются пять 32-битовых переменных.

A = 0x67452301

B = 0xEFCDAB89

C = 0x98BADCFE

D = 0x10325476

E = 0xC3D2E1F0

Вектор инициализации, подаваемый на вход 1-го раунда – результат конкатенации

SHA_0 = A||B||C||D||E

Определяются четыре нелинейные операции и четыре константы

F_t(m,l,k)=(m and l) or ((not m) and k)	K_t=0x5A827999	0 ≤ t ≤ 19
F_t(m,l,k)= m xor l xor k	K_t=0x6ED9EBA1	20 ≤ t ≤ 39
F_t(m,l,k)=(m and l) or (m and k) or (l and k)	K_t=0x8F1BBCDC	40 ≤ t ≤ 59
F_t(m,l,k)= m xor l xor k	K_t=0xCA62C1D6	60 ≤ t ≤ 79

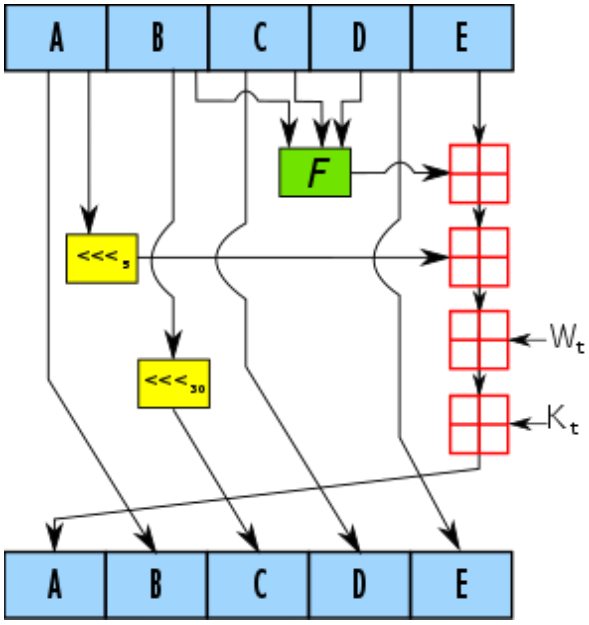


Рисунок 1.3 – Схема 1-го раунда SHA-1

Главный цикл

Главный цикл итеративно обрабатывает каждый 512-битный блок. В начале каждого цикла вводятся переменные a, b, c, d, e, которые инициализируются значениями A, B, C, D, E соответственно. Блок сообщения преобразуется из 16 32 битовых слов Mi в 80 32-битовых слов Wj по следующему правилу:

$W_t = M_t$ при 0 < = t< = 15

$W_t = (W_{t-3}$ xor W_{t-8} xor W_{t-14} xor $W_{t-16})<<1$ при 16 < = t< = 79,

где «<<» – это циклический сдвиг влево;

для t от 0 до 79

temp = (a<<5) + F_t(b,c,d) + e + W_t + K_t

e = d

d = c

c = b<<30

b = a

a = temp,

где «+»– сложение беззнаковых 32-битных целых чисел с отбрасыванием избытка (33-го бита).

После этого к A, B, C, D, E прибавляются значения a, b, c, d, e соответственно. Начинается следующая итерация.

Итоговым значением будет объединение пяти 32-битовых слов (A, B, C, D, E) в одно 160-битное хеш-значение.

Псевдокод SHA-1

Замечание: Все используемые переменные 32 бита.

Инициализация переменных:

h0 = 0x67452301

h1 = 0xEFCDAB89

h2 = 0x98BADCFE

h3 = 0x10325476

h4 = 0xC3D2E1F0

Предварительная обработка:

Присоединяем бит '1' к сообщению

Присоединяем k битов '0', где k наименьшее число ≥ 0 такое, что длина получившегося сообщения(в битах) сравнима по модулю 512 с 448.

Добавляем длину исходного сообщения (до предварительной обработки) как целое 64-битное big-endian число, в битах.

В процессе сообщение разбивается последовательно по 512 бит:

for перебираем все такие части

разбиваем этот кусок на 16 частей, слов по 32-бита (big-endian) w[i], 0 <= i <= 15

16 слов по 32-бита дополняются до 80 32-битовых слов:

for i **from** 16 **to** 79

w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) циклический сдвиг влево 1

Инициализация хеш-значений этой части:

a = h0

b = h1

c = h2

d = h3

e = h4

Основной цикл:
for i **from** 0 **to** 79
  if 0 ≤ i ≤ 19 **then**
    f = (b and c) or ((not b) and d)
    k = 0x5A827999
  else if 20 ≤ i ≤ 39 **then**
    f = b xor c xor d
    k = 0x6ED9EBA1
  else if 40 ≤ i ≤ 59 **then**
    f = (b and c) or (b and d) or (c and d)
    k = 0x8F1BBCDC
  else if 60 ≤ i ≤ 79 **then**
    f = b xor c xor d
    k = 0xCA62C1D6

  temp = (a leftrotate 5) + f + e + k + w[i]
  e = d
  d = c
  c = b leftrotate 30
  b = a
  a = temp

Добавляем хеш-значение этой части к результату:
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e

Итоговое хеш-значение(h0, h1, h2, h3, h4) должны быть преобразованы к big-endian):
digest = hash = h0 append h1 append h2 append h3 append h4

2 МАТЕМАТИЧЕСКИЕ ОСНОВЫ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

2.1 ОПРЕДЕЛЕНИЕ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Определение

Эллиптической кривой E над полем K называется множество точек $(x, y) \subset K \times K$, удовлетворяющих уравнению

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_i \in K \qquad (2.1)$$

вместе с точкой O – точка на бесконечности.

Вместо (2.1) используется и функция двух переменных

$$F(x,y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 - 0 \quad (2.2)$$

Введение операции сложения над парами точек E позволяет построить абелевую группу точек, если все точки E неособые (имеют однозначные производные). Такую кривую называют гладкой, или несингулярной.

Определение

Кривая E называется сингулярной (особой), если существует хотя бы одна точка *(x,y)*, в которой частные производные (2.2) одновременно обращаются в 0, т. е.

$$\partial F/\partial x = \partial F/\partial y = 0. \quad (2.3)$$

В противном случае кривая E называется несингулярной (неособой). Такие кривые представляют интерес для криптографии.

Вместо общей записи уравнения (2.1) часто рассматривают уравнения трех типов кривых:

$$E : y^2 = x^3 + ax + b, p \neq 2,3 \quad (2.4)$$

$$Es : y^2 + y = x^3 + ax + b, p = 2 \quad (2.5)$$

$$E_N : y^2 + xy = x^3 + ax^2 + b, p = 2, b \neq 0 \quad (2.6)$$

(2.4) описывает все кривые над полями характеристики, не равной 2 и 3, (2.5) и (2.6) – кривые над полями характеристики 2.

Несингулярная кривая (2.4), таким образом, не имеет кратных корней кубического трехчлена *f(x)* (т. е. кубическое уравнение имеет три различных вещественных корня либо один вещественный и два комплексно-сопряженных корня).

Для несингулярной кривой (2.4) должно выполняться условие:

$$\Delta = (4a^3 + 27b^2) \neq 0 \quad (2.7)$$

Величину Δ называют дискриминантом кубического трехчлена.

2.2 ЗАКОНЫ СЛОЖЕНИЯ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ И ПОСТРОЕНИЕ АБЕЛЕВОЙ ГРУППЫ ТОЧЕК

Перед обсуждением конкретных примеров эллиптических кривых над различными полями отметим чрезвычайно важное свойство точек эллиптической кривой: они образуют абелеву группу относительно операции сложения точек, о которой будет подробнее сказано ниже. Чтобы объяснить наглядно, как это получается, временно будем полагать, что K = R. т.е. что эллиптическая кривая – обычная плоская кривая (с добавлением еще одной точки O «в бесконечности»).

Определение

Пусть E – эллиптическая кривая над вещественными числами, и пусть P и Q – две точки на E. Определим точки -P и P+Q по следующим правилам:

1.Точка O – тождественный элемент по сложению («нулевой элемент») группы точек. В следующих пунктах предполагается, что ни P, ни Q не являются точками в бесконечности.

2. Точки P = (x, y) и – P имеют одинаковые x-координаты, а их y-координаты различаются только знаком, т. е. − (x, y) = (x,−y). Из (1) сразу следует, что (x,−y) – также точка на E.

$$-P = (x_1, -y_1) \tag{2.8}$$

3. Если P и Q имеют различные x - координаты, то прямая L = PQ имеет с E еще в точности одну точку пересечения R (за исключением двух случаев: когда она оказывается касательной в P, и тогда полагаем R = P, или касательной в Q, и тогда полагаем R =Q). Определяем теперь P + Q как точку – R, т. е. как отражение от оси *x* третьей точки пересечения. Геометрическое построение, дающее P + Q, приводится ниже в примере 1.

4. Если Q = – P (т. е. *x* – координата Q та же, что и *y* P, а y-координата отличается лишь знаком), то полагаем P + Q = O («точке в бесконечности»; это является следствием правила 1).

5. Остается возможность P = Q. Тогда считаем, что L – касательная к кривой в точке P. Пусть R – единственная другая точка пересечения L с E. Полагаем

P + Q = – R (в качестве R берем P, если касательная прямая в P имеет «двойное касание», т. е. если P есть точка перегиба кривой).

Пример 1

На рисунке 2.1 слева изображена эллиптическая кривая $y^2 = x^3 - x$ в плоскости xOy и приведен типичный случай сложения точек P и Q. Чтобы найти P + Q, проводим прямую PQ и в качестве P + Q берем точку, симметричную относительно оси x третьей точке, определяемой пересечением прямой PQ и кривой. Если бы P совпадала с Q, т. е. если бы нам нужно было найти 2P, то использовали бы касательную к кривой в P : тогда точка 2P симметрична третьей точке, в которой эта касательная пересекает кривую.

На рисунке 2.1 справа аналогичным образом проиллюстрировано сложение точек R и Q на кривой $y^2 = x^3 + x + 1$.

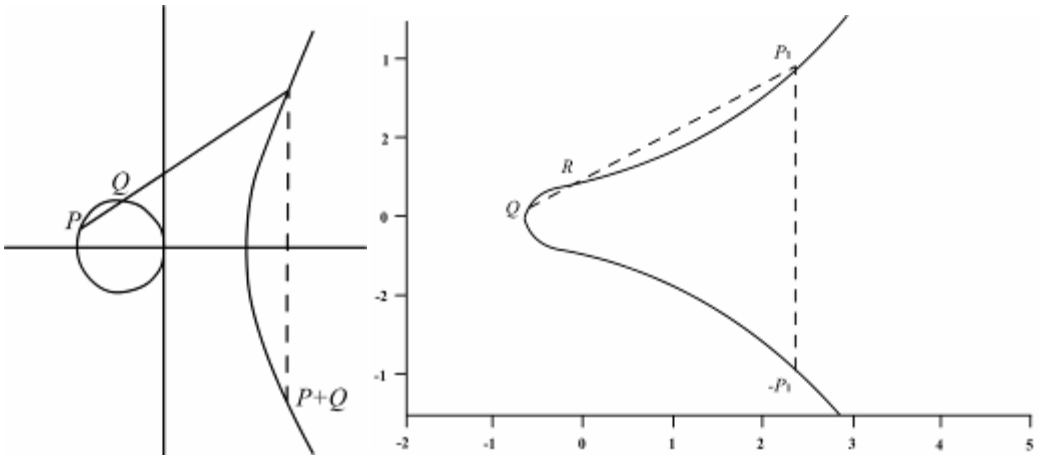
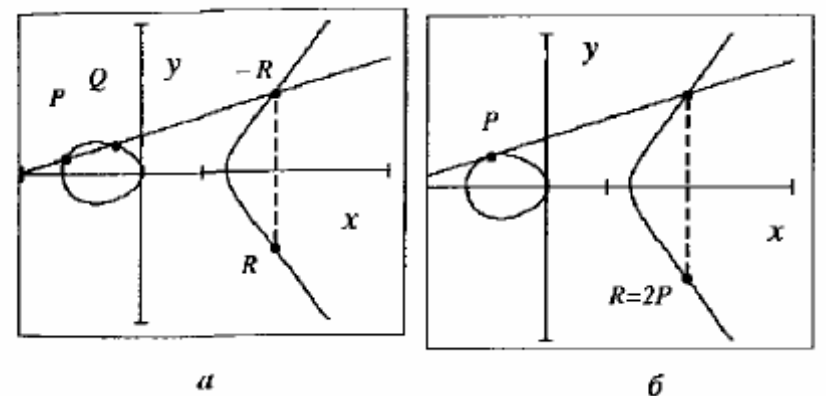


Рисунок 2.1 – Примеры геометрического построения суммы точек эллиптической кривой

Обозначим $(x_1,y_1),(x_2,y_2)$ и (x_3,y_3) – координаты точек P, Q и P+Q соответственно. Необходимо выразить (x_3,y_3) через $(x_1,y_1),(x_2,y_2)$.

Имеют место два случая:



а *б*

Рисунок 2.2 – Примеры геометрического построения суммы точек эллиптической кривой

1. P ≠ ±Q. Уравнение прямой линии, проходящей через точки P и Q (рисунок 2.1а), имеет вид

$$y = \lambda x + \beta;\ \lambda = \frac{y_2 - y_1}{x_2 - x_1};\ \beta = y_1 - \lambda x_1; \tag{2.9}$$

Уравнение (2.2) в канонической форме (2.4) можно переписать

$$F(x,y) = y^2 - x^3 - ax - b = 0. \tag{2.10}$$

Точки пересечения кривой E и прямой (2.2) имеют по оси x координаты x_1, x_2, x_3 точек P, Q и $-$ R соответственно. Поскольку они являются общими для функций (2.9) и (2.10), последнее уравнение можно записать в виде

$(\lambda x = \beta)^2 - x^3 - ax - b = 0$, или $-(x - x_1)(x - x_2)(x - x_3) = 0.$

Приравнивая в этих кубических уравнениях коэффициенты при переменных x_2, получим:

$$\lambda^2 = x_1 + x_2 + x_3. \tag{2.11}$$

Параметр λ прямой (2.9) можно также выразить в виде

$$\lambda = \frac{-y_3 - y_1}{x_3 - x_1}.$$

Из (2.11) и последнего соотношения окончательно имеем координаты точки

R=P+Q= (x_3, y_3);

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = -y_1 - \lambda(x_3 - x_1), \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \end{cases}. \qquad (2.12)$$

2. P = Q, R = 2P. В этом случае $x_1 = x_2$ и параметр λ не определен. Дифференциал функции (2.4)

$$2ydy = 3x^2dx + adx;$$

тогда при $x = x_1$ производная равна параметру ν касательной $y = \nu x + \beta$ к кривой в точке P

$$\nu = \left.\frac{dy}{dx}\right|_{x=x_1} = \frac{3x_1^2 + a}{2y_1}.$$

Теперь можно записать координаты точки R = 2P = (x_3, y_3):

$$\begin{cases} x_3 = \nu - 2x_1, P = Q; \\ y_3 = -y_1 - \nu(x_3 - x_1), \nu = \dfrac{3x_1^2 + a}{2y_1}. \end{cases} \qquad (2.13)$$

Если n – целое число, то, как и в любой абелевой группе, nP обозначает сумму n точек P при n > 0 и сумму |n| точек – P, если n ≤ 0 .

Формулы сложения (2.12) и удвоения (2.13) справедливы для кривых E над всеми полями, в том числе и конечными, кроме полей характеристик 2 и 3. В последнем случае, как видно из (2.13), редукция по модулю 2 или 3 ведет к некорректности формул удвоения и следует использовать другие канонические уравнения кривых. Заметим, что координаты сложения и удвоения точек определяются с помощью всех операций в поле, т. е. сложения (вычитания), умножения и деления.

Для построения абелевой группы точек E определим O группы как

P+(-P) = O, ∀ P∈ E.

Если провести прямую через точки P и – P, то третья точка пересечения прямой и E уходит в бесконечную точку вдоль оси y . Поэтому O группы точек E называют «точкой на бесконечности».

Смысл перехода к обратной к точке пересечения прямой и кривой E при определении суммы R = P + Q становится понятным, если выразить, например, точку P как P = R – Q. В этом случае прямая проходит через точки R, – Q и – P, а обратной к этой третьей точке является точка P. Для точек E выполняется ассоциативность сложения

P+(Q+S) = (P+Q)+S и коммутативность P+Q=Q+P.

Таким образом, множество точек E замкнуто относительно операции сложения, удовлетворяет свойствам ассоциативности, коммутативности, имеет обратный элемент и O (нуль группы), т. е. удовлетворяет всем условиям аддитивной абелевой группы.

Пример 2

Пусть P = (–3, 9) и Q = (–2, 8) – точки на эллиптической кривой $y_2 = x_3 - 36x$. Найти P + Q и 2P.

Решение:

Подстановка $x_1 = -3, y_1 = 9, x_2 = -2, y_2 = 8$ в первое из уравнений (2.5) дает $x_3 = 6$; тогда второе из уравнений (2.5) дает $y_3 = 0$. Непосредственной подстановкой координат точки P + Q = (6, 0) в уравнение кривой можно убедиться в том, что она также лежит на ней.

Далее, подставляя $x_1 = -3, y_1 = 9, a = -36$ в первое из уравнений (2.6), получаем для x-координаты точки 2P значение 25/4, а второе из уравнений (2.6) дает для y-координаты значение –35/8. Точка 2P = (25/4, –35/8) также принадлежит рассматриваемой кривой.

2.3 ПОРЯДОК ГРУППЫ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ И ПОРЯДОК ТОЧКИ

Аналогией с экспоненцированием в мультипликативной группе в аддитивной группе точек является k-кратное сложение элементов (в нашем случае – точек P), которое обозначается как P + P + P + ... + P = kP.

Точку kP называют скалярным произведением, а процесс ее вычисления экспоненцированием точки P (возведением в степень).

Определение

Порядком N_E эллиптической кривой называется число всех ее точек (x, y) вместе с точкой на бесконечности (точкой O).

Определение

Порядком точки P эллиптической кривой называется наименьшее натуральное число $m \neq 0$, для которого mP = O.

Порядки N_E и m могут быть бесконечными и конечными. В группе бесконечного порядка (например, в группе точек E над полями R или Q) могут быть точки конечного порядка. В частности, в группе точек E над R всегда есть точки 2-го и 3-го порядков.

Координаты x_i точек 2-го порядка – это корни кубического трехчлена правой части (2.4). Для кубического трехчлена с тремя действительными корнями a_1, a_2, a_3 (рисунок 2.2 а) имеем три точки второго порядка $(a_1,0),(a_2,0),(a_3,0)$. В этих точках $(a_i,0) = -(a_i,0) \Rightarrow 2(a_i,0) = O$.

Вместе с точкой O эти точки образуют подгруппу 4-го порядка группы точек E бесконечного порядка. Кубический трехчлен с одним действительным корнем (рисунок 2.2б) порождает подгруппу 2-го порядка точек 2-го порядка.

Точки 3-го порядка в группе точек E над R можно найти из соотношений

2P = –P => 3P = O => $x_3 = x_1$. Согласно (2.13)

$$\left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 = x_1 \Rightarrow x_1 \geq 0.$$

Отсюда с учетом (2.4)

$$3x_1^4 + 6ax_1^2 + 12bx_1 - a^2 = 0, x_1 \geq 0. \quad (2.14)$$

В частности, при a = 0

$$x_1(x_1^3 + 4b) = 0 \Rightarrow x_1 = 0.$$

В общем случае x-координата точки третьего порядка совпадает с x-координатой точки перегиба эллиптической кривой E.

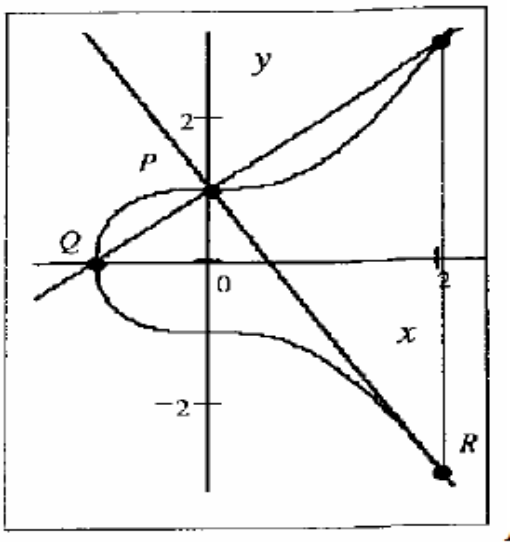


Рисунок 2.3 – График кривой $y^2 = x^3 + 1$

Точки конечного порядка кривой образуют так называемые подгруппы кручения. На кривой $y^2 = x^3 + 1$ на рисунке 3.1 имеются точки 2, 3 и 6-го порядков, образующие циклические подгруппы кручения тех же порядков.

Точка Q – точка 2-го порядка, являющаяся генератором подгруппы кручения G_2 = {O, Q} 2-го порядка. Точка P – точка 3-го порядка, генерирующая подгруппу G_3= {O, P, 2P} того же порядка. Точка R – точка 6-го порядка, образующая подгруппу G_6 = {O, R, 2R = 2P = -P, 3R = Q, 4R = P, 5R = -R} порядка 6.

Результат 2R = 2P следует из удвоения суммы P + Q= R. Очевидно также, что группа G_6, может быть выражена через прямую сумму циклических подгрупп 2-го и 3-го порядков

$$G_6 = G_2 \otimes G_3 = \{0, Q\} \otimes \{0, P, 2P\}.$$

Таким образом, любая точка группы G_6 выражается через сумму точек из подгрупп G_2 и G_3.

Наряду с циклическими подгруппами в группах точек E встречаются нециклические подгруппы кручения.

Например, уравнение $y^2 = (x - a_1)(x - a_2)(x - a_3)$ над полем R с тремя вещественными корнями порождает 3 точки 2-го порядка и соответствующие нециклические подгруппы кручения (нециклическая подгруппа 4-го порядка, из 3 циклических подгрупп (точек 2-го порядка) кручения; нет генерирующей точки для этой подгруппы).

Пример 3

Найти порядок точки P = (2, 3) на $y^2 = x^3 + 1$.

Решение:

Применяя (2.6), находим, что 2P = (0, 1), и вновь, с помощью (2.6), что 4P = 2(2P) = (0, −1). Поэтому 4P = −2P и, следовательно, 6P = O. Тем самым порядок P может быть равен 2, 3 или 6. Но 2P = (0,1) ≠ O, а если бы P имела порядок 3, то было бы 4P = P, что неверно. Итак, P имеет порядок 6.

2.4 ПРОЕКТИВНЫЕ КООРДИНАТЫ

При рассмотрении E полезным оказывается переход от аффинных координат (x,y) к проективным (X, У, Z), связывающий точки кривой E в этих координатах отношением эквивалентности. Привлечение новой переменной Z позволяет задать координаты нуля группы E (точки на бесконечности). В операциях над конечными полями проективные координаты позволяют избежать трудоёмких вычислений обратного элемента поля при сложении точек.

Определение

Проективной плоскостью над полем K называется множество классов эквивалентности троек (X,Y,Z), в которых хотя бы один элемент ненулевой. Эквивалентными считаются тройки, если элементы одной из них получаются из другой умножением на скаляр: (X',Y',Z')~(X,Y,Z), если для некоторого элемента $\lambda \in K : (\lambda X', \lambda Y', \lambda Z') = (X, Y, Z)$.

Такие классы эквивалентности называются проективными точками. Например, две точки (7, 1, 1) и (8, 3, 3) эквивалентны в проективной плоскости над F_{13} (λ =3). Проективные точки с ненулевым элементом Z принадлежат классу эквивалентности, содержащему единственную точку, вида $(x, y, 1)$: она просто вычисляется x = X/Z, y = Y/Z.

В аффинных координатах уравнение кривой (1.4) запишем как

$$F(x,y) = \left(\frac{Y}{Z}\right)^2 - \left(\frac{X}{Z}\right)^3 - a\left(\frac{X}{Z}\right) - b = 0.$$

Умножим данное уравнение на Z_3

$$F(X,Y,Z) = Z^3 F(x,y) = Y^2 Z - X^3 - aXZ^2 - bZ^3 = 0. \qquad (2.15)$$

Исключая из рассмотрения начало координат (0, 0, 0), для любой тройки (X,Y,Z) класс эквивалентности задается проективной точкой (λX, λY, λZ), где λ – скаляр, X, Y, Z – фиксированы. В трехмерном пространстве этот класс представляет собой прямую линию, проходящую через начало координат. При Z ≠ 0 любая такая прямая пересекает плоскость Z = 1, в которой, как видно из (4.1), возвращаемся к записи кривой в аффинных координатах. Таким образом, проективная плоскость может быть определена как множество всех точек (x, y) обычной (аффинной) плоскости с дополнением точек, для которых Z = 0.

Кроме точек с Z ≠ 0 уравнению (2.15) удовлетворяет еще одна точка.

Подставим Z = 0 в уравнение, получим $X_3 = 0$, это означает, что $X = 0$. Но имеется только один класс эквивалентности, где оба элемента X и Z нулевые – класс, содержащий (0,1,0). В проективной плоскости она задает координаты бесконечно удаленной точки или нулевого элемента группы точек E. Точка O является третьей точкой пересечения точек P и −P на бесконечности.

2.5 ДИСКРИМИНАНТ И J-ИНВАРИАНТ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

При изучении многих свойств и преобразований эллиптических кривых часто полезными оказываются дискриминант Δ и j-инвариант кривой. В частности, условие Δ ≠ 0 является необходимым и достаточным условием несингулярности кривой над любым полем.

Для кривой (2.4) дискриминант кубического уравнения

$$x^3 + ax + b = (x - e_1)(x - e_2)(x - e_3) = 0$$

определен формулой (2.7).

Найдем корни кубического трехчлена. Будем искать решение в форме

$$x = \sqrt[3]{r} + \sqrt[3]{s},$$

при этом

$$x^3 = r + s + 3\sqrt[3]{rs}(\sqrt[3]{r} + \sqrt[3]{s}) = r + s + 3x\sqrt[3]{rs}.$$

Сравнивая это уравнение с исходным, получаем $3\sqrt[3]{rs} = -a, r + s = -b$ и, следовательно,

$$rs = -\frac{a^3}{27}; r + s = -b \Rightarrow r^2 + br - \frac{a^3}{27} = 0.$$

Решение этого квадратного уравнения дает пару значений

$$r, s = -\frac{b}{2} \pm \frac{1}{6}\sqrt{\frac{-\Delta}{3}}; \Delta = -(4a^3 + 27b^2).$$

Дискриминант Δ кубического трехчлена совпадает с дискриминантом квадратного уравнения, определяющего значения r, s. Корни кубического уравнения теперь можно найти из

$$e = x = \sqrt[3]{r} + \sqrt[3]{s} = \sqrt[3]{r} - \frac{a}{3\sqrt[3]{r}}. \qquad (2.16)$$

В общем случае дискриминант полинома *n*-й степени

$$f(x) = x^n + a_{n-1}x^{n-1} + ... + a_1x + a_0 = (x - e_1)(x - e_2)(x - e_3)...(x - e_n)$$

определяется как произведение квадратов разностей всех корней

$$\Delta = \prod_{i>k}(e_i - e_k)^2. \qquad (2.17)$$

Для кривой (2.1) общего вида определим вспомогательные коэффициенты c_{2i}, d_4, дискриминант Δ и j-инвариант равенствами:

$$c_2 = a_1^2 + 4a_2;$$

$$c_4 = 2a_4 + a_1 a_3;$$

$$c_6 = a_3^2 + 4a_6;$$

$$c_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2;$$

$$d_4 = c_2^2 - 24c_4;$$

$$\Delta = -c_2^2 c_8 - 8c_4^3 - 27c_6^2 + 9c_2 c_4 c_6;$$

$$j = \frac{d_4^3}{\Delta}, \Delta \neq 0.$$

В частности, для кривой (2.4) получим

$$\Delta = -16 \bullet (4a^3 + 27b^2); j(E) = \frac{-4^3 12^3 a^3}{\Delta} = \frac{12^3 4a^3}{4a^3 + 27b^2}, \Delta \neq 0. \quad (2.18)$$

Отсюда видно, что кривые с коэффициентом a = 0 – кривые с нулевым j-инвариантом, а кривые с коэффициентом b = 0 – кривые с j-инвариантом, равным $12^3 = 1728$.

Для кривых (2.5), (2.6) характеристики 2 соответственно имеем

$$E_S : \Delta = 1, j = 0; \quad (2.19)$$

$$E_N : \Delta = b, j = 1/\Delta = b^{-1}. \quad (2.20)$$

Изоморфные кривые и так называемые кривые кручения имеют один и тот же j-инвариант. Кривые с нулевым j-инвариантом над некоторыми полями не рекомендуются для криптографических применений.

Можно заметить, что всегда можно найти кривую с заданным $j(E) = j_0$.

Пусть $a = 3k(\mathrm{mod}\, p), b = 2k(\mathrm{mod}\, p)$, тогда $j_0 = 1728\frac{k}{k+1} \mathrm{mod}\, p$ и $k = \frac{j_0}{1728 - j_0} \mathrm{mod}\, p.$

2.6 ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ НАД ПРОСТЫМИ ПОЛЯМИ ГАЛУА

Рекомендуемыми в появившихся за последние годы стандартах являются два типа полей – простые поля Галуа F_p,и расширенные поля F_2^m характеристики 2.

Рассмотрим свойства кривых E над простыми полями F_P, p > 3. В ряде случае операции над простым полем оказываются менее сложными (и более быстрыми), чем над расширенным полем.

Пусть кривая (2.4) с целыми коэффициентами a и b определена над полем рациональных чисел и p > 3 – простое число. Тогда сравнение

$$y^2 = x^3 + ax + b(\mathrm{mod}\, p) \tag{2.21}$$

называется редукцией кривой по модулю p. При этом и коэффициенты кривой и координаты (x, y) точек являются целыми сравнимыми по модулю p числами.

Редукция называется хорошей, если p не делит дискриминант или

$$\Delta = -(4a^3 + 27b^2)\,\mathrm{mod}\, p \neq 0. \tag{2.22}$$

В этом случае все корни кубического уравнения различны и кривая является несингулярной (без особых точек). Будем рассматривать только такие кривые.

Редукция (2.21) равнозначна переходу от поля Q к конечному полю, в частности, простому полю Галуа, т. е.

$$y^2 = x^3 + ax + b; a, b \in F_p. \tag{2.23}$$

Абелеву группу точек (x, y), удовлетворяющих уравнению (2.23), вместе с точкой на бесконечности O обозначим E_p. В отличие от коэффициентов a, b кривой координаты (x, y) точек могут рассматриваться как элементы любого расширения $F_p^m (m = 1,2,3,...)$ поля F_p вплоть до алгебраического замыкания $\overline{F}_p$.

Законы сложения точек (2.12), (2.13) справедливы для группы E_p ($p \neq 2, 3$) после введения редукции по модулю p, а операция деления равнозначна умножению на обратный элемент поля F_p. Из конечности числа элементов поля, очевидно, следует и конечность числа точек кривой, т. е. ее порядка.

Так как кривая всегда содержит точку на бесконечности O, а для каждого решения *x* уравнения (2.23) имеются по два значения y, для числа N_E точек кривой можно получить грубую оценку

$$1 < N_E < 2p + 1, \text{или} \left| p + 1 - N_E \right| < p.$$

Более точную оценку порядка N_E эллиптической кривой E над конечным полем F_q получил в 1934 г. немецкий математик *Г. Хассе.*

Теорема (*Хассе*)

Для эллиптической кривой E над конечным полем F_q справедлива следующая оценка порядка кривой N_E

$$q + 1 - 2\sqrt{q} \le N_E \le q + 1 + 2\sqrt{q}, q = p^m, m = 1,2,3,... . \tag{2.24}$$

В частности, для простого поля она может быть записана как

$$\left| p + 1 - N_E \right| < 2\sqrt{p}\,. \tag{2.25}$$

Пусть $\chi(z)$ – квадратичный характер элемента z поля F_q, определяемый как

$$\chi(z)=\begin{cases}1, z=a^2, a\in F_q,\\ -1, z\neq a^2,\\ 0, z=0,\end{cases} \qquad (2.26)$$

Иными словами, если z имеет корень квадратный в поле F_q, $\chi(z)$ = 1, в противном случае $\chi(z)$ = – 1. В первом случае говорят, что z является квадратичным вычетом, во втором квадратичным невычетом. Тогда с учетом (2.23) порядок кривой можно рассчитать перебором всех элементов поля F_q как сумму

$$\sum_{x\in F_q}\{\chi(x^3+ax+b)+1\}=q+1+\sum_{x\in F_q}\chi(x^3+ax+b)=q+1-t, \qquad (2.27)$$

где

$$t=-\sum_{x\in F_q}\chi(x^3+ax+b).$$

Первая единица в выражении (2.27) учитывает точку на бесконечности O, а под знаком суммы каждое решение *x* уравнения (2.23) даст по две точки E.

Исключением являются точки второго порядка с координатой y = 0, которые в соответствии с (2.26), (2.27) учитываются по одному разу. Значение t в (2.27), не превышающее по абсолютной величине $2\sqrt{q}$, может быть положительным или отрицательным в зависимости от преобладания квадратичных вычетов или невычетов $f(x)=x^3+ax+b$.

2.7 МЕТОДЫ ЭКСПОНЕНЦИРОВАНИЯ ТОЧКИ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Наиболее распространенной операцией во всех криптографических алгоритмах является *k*-кратное сложение точки *P*, обозначаемое как kP. Эту операцию обычно называют скалярным умножением, или, в терминологии мультипликативной группы, экспоненцированием точки кривой.

Дадим краткое описание методов повышения производительности при вычислении точки kP. Подход к расчету точки kP может отличаться в зависимости от того, является ли точка P фиксированной (заранее известной) или произвольной точкой. В первом случае всегда можно пользоваться предвычислениями точек, например 2^iP, которые хранятся в памяти. Двоичное представление числа *k* позволяет селектировать те из них, которые в результате суммирования образуют точку *kP*. Во втором, более общем случае, все вычисления приходится проводить в реальном времени.

Пусть порядок $Ord\ P=r, [\log_2 r]=L$ и число *k* представлено в двоичной системе $k=\sum_{i=0}^{L-1}k_i 2^i$.

3 ЭЛЕКТРОННО-ЦИФРОВАЯ ПОДПИСЬ НА БАЗЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Электронная цифровая подпись (ЭЦП) используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает ее основными достоинствами:

– удостоверяет, что подписанный текст исходит от лица, поставившего подпись;

– не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;

– гарантирует целостность подписанного текста.

Цифровая подпись представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Система ЭЦП включает две процедуры:

1) процедуру постановки подписи;

2) процедуру проверки подписи.

В процедуре постановки подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи – открытый ключ отправителя.

При формировании ЭЦП отправитель прежде всего вычисляет хеш-функцию h(M) подписываемого текста M. Вычисленное значение хеш-функции h(M) представляет собой один короткий блок информации m, характеризующий весь текст M в целом. Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

При проверке ЭЦП получатель сообщения снова вычисляет новое значение хеш-функции m = h(M) принятого по каналу текста M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хеш-функции.

То есть выполняется аутентификация автора документа и самого документа, установления подлинности автора и отсутствия изменений в полученном документе.

Аутентификация (англ. authentication) - процедура проверки подлинности, например:

– проверка подлинности пользователя путем сравнения введённого им пароля (для указанного логина) с паролем, сохранённым в базе данных пользовательских логинов;

– подтверждение подлинности электронного письма путем проверки цифровой подписи письма по открытому ключу отправителя;

– проверка контрольной суммы файла на соответствие сумме, заявленной автором этого файла.

3.1 АЛГОРИТМ ECDSA

Национальный стандарт DSS правительства США принятый NIST сменил предыдущий стандарт FIPS 186-1, действующий с 1994 года, и рекомендует ECDSA взамен DSA, построенного на арифметике простого поля Галуа.

Описание ECDSA

Параметры пользователя

Пользователь A:

– генерирует и хранит в секрете долговременный секретный ключ как целое число $0 < d_A < n$;

– вычисляет открытый ключ как точку кривой $Q_A = d_A G$. Открытый ключ доступен для всех пользователей системы.

Формирование ЭЦП

Пользователь А:

1. Вычисляет хеш значение сообщения *M* как целое число $e = h(M), e < n$.

2. Генерирует случайное целое число $0 < k_A < n$.

3. Вычисляет точку $R = k_A G = (x_1, y_1)$.

4. Вычисляет параметр $r = \pi(R) \bmod n$. При $r = 0$ возврат в пункт 2.

5. Вычисляет обратный элемент k_A^{-1} простого поля F_n.

6. Вычисляет параметр $s = k_A^{-1}(e + d_A r) \bmod n$. При $s = 0$ возврат в пункт 2.

7. Направляет пользователю В подписанное сообщение *(M, r, s)*, в котором

DS = (r, s) – цифровая подпись.

В пункте 4 преобразование точки *R* в целое число предполагает, что ее *x* – координата как элемент x_1 поля F_n переводится в целое число $\overline{x_1}$ с последующей редукцией по модулю $n (r = \pi(R) \bmod n = \overline{x_1} \bmod n)$.

Проверка ЭЦП

Пользователь В проверяет ЭЦП пользователя А, имея в распоряжении следующую информацию: открытый ключ пользователя А Q_A, общесистемные параметры, алгоритм хеширования *h(M)* и подписанное сообщение *(M, r, s).* Суть проверки состоит в вычислении на основе известных данных параметра *r'* и сравнении его с принятым значением *r.*

Умножив равенство в пункте 6 на инверсию s^{-1} второго параметра подписи и учитывая, что $Q_A = d_A G$ для точек криптосистемы в результате экспоненцирования (скалярного произведения), получим равенство

$k_A G = s^{-1} e G + s^{-1} r d_A G = uG + vQ_A$;

$u = s^{-1} e \bmod n$;

$v = s^{-1} r \bmod n$.

Согласно пунктам 3 и 4 протокола формирования левая часть этого равенства определяет точку $R = (x_1, y_1)$ и, соответственно, параметр $r = \overline{x_1} \bmod n$.

Правая часть равенства включает известные получателю данные, которые он использует для вычисления параметра *r'* (он может оказаться отличным от параметра r при модификациях сообщения *M* и ошибках в канале связи).

Протокол проверки ЭЦП включает следующие вычисления

Пользователь В:

1. Вычисляет хеш-значение полученного сообщения $M: e = h(M), e < n$.

2. Вычисляет обратный элемент $s^{-1} \bmod n$ поля F_n.

3. Вычисляет параметры $u = s^{-1} e \bmod n, v = s^{-1} r \bmod n$.

4. Вычисляет точку $R' = uG + vQ_A = (x_1, y_1)$.

5. Вычисляет параметр $r' = \pi(R') = \overline{x_1}' \bmod n$.

6. Сравнивает вычисленное *r '* и принятое значения *r.* При равенстве *r'* = *r* цифровая подпись верна, в противном случае она отвергается.

В результате проверки пользователь В удостоверяется в подлинности отправителя А и целостности сообщения М.

В ряде проектов и стандартов определение параметра *r* подписи не регламентируется, а задается функцией $r = \pi(x_1, y_1) \bmod n$ Это, в частности, позволяет избежать неоднозначности определения $r = \overline{x_1} \bmod n$ в связи с наличием обратной точки $(n - k_A)G = -k_A G$, имеющей ту же *x*-координату, что и точка $k_A G$ (и, следовательно, совпадающий параметр *r*).

FIPS 186-2-2000 рекомендует к использованию 10 полей и 15 эллиптических кривых, 5 из которых определены над простыми полями F_p и 10 – над расширенными полями F_2^m. Значения модулей простых полей:

$$P_{192} = 2^{192} - 2^{64} - 1;$$

$$P_{224} = 2^{224} - 2^{96} + 1;$$

$$P_{256} = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1;$$

$$P_{384} = 2^{384} - 2^{128} + 2^{96} + 2^{32} - 1;$$

$$P_{521} = 2^{521} - 1.$$

Расширения двоичного поля равны *m = 163, 233,283,409* и *571*.

Из кривых над простыми полями F_P *NIST* рекомендует кривую

$$y^2 = x^3 - 3x + b \bmod p.$$

Значения коэффициента *b* для пяти рекомендуемых кривых, координаты одного из возможных генераторов - точки $G = (G_x, G_y)$ порядка *n* приведены в стандарте. Из 10 кривых над расширенными полями F_2^m 5 несуперсингулярных кривых

$y^2 + xy = x^3 + ax^2 + b$ с коэффициентами $a = 1$ и $b \in F_2^m, b \neq 0,1$.

Еще 5 несуперсингулярных кривых вида $y^2 + xy = x^3 + ax^2 + 1$ над полем F_2^m с коэффициентами *a =1* или *0*. Эти кривые называют кривыми Коблица, в стандарте они обозначены как $(K - m)$. В стандарте *FIPS 186-2-2000*, кроме порядков кривых, даны возможные значения координат точек *G* порядка *n*.

Кривые Коблица - наиболее технологичные кривые над полем характеристики 2, они обеспечивают наивысшую производительность вычислений в поле F_2^m В то же время они относятся к классу аномальных кривых, что снижает их стойкость в $\sqrt{m}$ раз по

сравнению с кривыми $(B-m)$ с произвольным значением коэффициента *b* и таким же порядком.

Достаточно большой диапазон размеров ноля и порядков криптосистем позволяет реализовать системы с различной степенью безопасности, работающие совместно с симметричными блочными шифрами с разной длиной ключа. Однако с ростом размера модуля падает скорость криптопреобразований и, соответственно, растет время шифрования и обмена ключами, формирования и проверки подписи.

В целом, при одинаковых алгоритмах и программной реализации арифметика эллиптических кривых над простым полем F_p выполняется в 2-3 раза быстрее, чем в поле F_2^m.

4 ПРОТОКОЛЫ СЛЕПОЙ ПОДПИСИ

4.1 ПОЛНОСТЬЮ СЛЕПАЯ ПОДПИСЬ

Дана ситуация: Боб – нотариус. Алисе нужно, чтобы он подписал документ, не имея никакого представления о его содержании. Боб только заверяет, что документ нотариально засвидетельствован в указанное время. Тогда они действуют по следующему алгоритму:

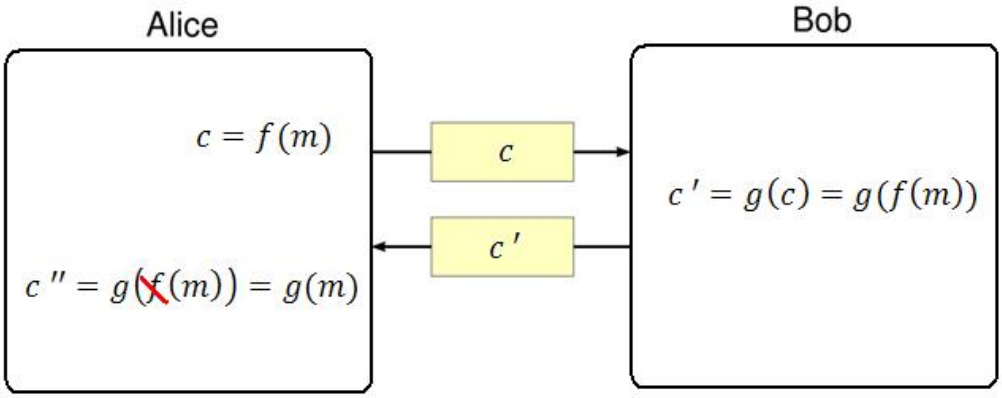


Рисунок 4.1 – Схема полностью слепой подписи

В этой схеме Алиса хочет, чтобы Боб вслепую подписал сообщение m. Для этого:

1. Алиса зашифровывает сообщение *m* функцией *f,* получая зашифрованное сообщение *c = f(m).*
2. Алиса отсылает зашифрованное сообщение Бобу.
3. Боб вслепую (так как не знает, что находится внутри) подписывает сообщение с функцией *g*, получая *c' = g(c) = g(f(m)).*
4. Боб посылает *c'* обратно Алисе.
5. Алиса получает *c'* и убирает своё шифрование, получая: *c" = g(f(m))*f⁻¹ = g(m).*

Этот протокол работает, только если функции подписи и шифрования коммутативны.

4.2 СЛЕПАЯ ПОДПИСЬ

1. Боб готовит *n* документов, на каждом из которых написано некоторое уникальное слово (чем больше *n*, тем меньше у Боба шансов смошенничать).

2. Боб маскирует каждый документ уникальным маскирующим множителем и отправляет их Алисе.

3. Алиса получает все документы и случайным образом выбирает *n-1* из них.

4. Алиса просит Боба выслать маскирующие множители для выбранных документов.

5. Боб делает это.

6. Алиса вскрывает *n-1* документов и убеждается, что они корректны.

7. Алиса подписывает оставшийся документ и отсылает Бобу.

8. Теперь у Боба есть подписанный Алисой документ с уникальным словом, которое Алиса не знает.

4.3 ПРОТОКОЛ RSA

Первая реализация слепых подписей была осуществлена Чаумом с помощью криптосистемы RSA:

Допустим, что изначально у Боба есть открытый ключ *(p,e),* где *p* – это модуль, а *e* – публичная экспонента ключа.

1. Алиса выбирает случайный маскирующий множитель *r*, взаимно простой с *p*, и вычисляет $m' \equiv mr^e \pmod p$.

2. Алиса посылает *m'* по открытому каналу Бобу.

3. Боб вычисляет $s' \equiv (m')d \pmod p$, используя свой закрытый ключ *(p, d).*

4. Боб отсылает *s'* обратно Алисе.

5. Алиса убирает свою изначальную маскировку и получает подписанное Бобом исходное сообщение *m* следующим образом: $s \equiv s' * r^{-1} \pmod p \equiv m^d \pmod p$.

6. Для проверки подписи Алисе необходимо возвести подписанное Бобом сообщение в степень *e.* Если полученное сообщение совпадает с тем, что она отправила, подпись корректна.

Чаум придумал целое семейство более сложных алгоритмов слепой подписи под общим названием неожиданные слепые подписи.

4.4 СЛЕПАЯ ПОДПИСЬ НА ОСНОВЕ ЭЦП ШНОРРА

Пусть Алиса хочет подписать сообщение *m* у Боба таким образом, чтобы, во-первых, Боб не мог ознакомиться с сообщением в ходе подписи, во-вторых, чтобы Боб не мог впоследствии при получении сообщения *m* и соответствующей подписи идентифицировать пользователя, инициировавшего протокол слепой подписи для данного конкретного сообщения (Алису). Данный протокол реализуется следующим образом:

1. Алиса инициирует взаимодействие с Бобом.

2. Боб отправляет Алисе значение $R = a^k \bmod p$.

3. Алиса вычисляет значения $R' = Ra^{-w}y^{-t} \pmod y$, где ($w$ и t – случайные числа, не превосходящие y), $E' = H(m||R')$ и $E = E'+t \pmod y$, после чего отправляет Бобу значение E.

4. Боб вычисляет значение S, такое, что $R = a^S y^E \bmod p$, и отправляет S Алисе.

5. Алиса вычисляет подпись *(E', S'),* где $E' = E^{-t} \bmod y$ и $S' = S - w \pmod y$, которая является подлинной по отношению к сообщению *m.*

4.5 СЛЕПАЯ ПОДПИСЬ НА ОСНОВЕ ГОСТ Р 34.10-94

Параметры

p – простое число, $510 \le |p| \le 512$ (либо $1022 \le |p| \le 1024$), где $|p|$ – разрядность двоичного представления числа p.

q – большой простой делитель числа $p-1$, $255 \le |q| \le 256$ (либо $511 \le |q| \le 512$)

$\alpha \ne 1$, $\alpha < p$, $\alpha^q \bmod p = 1$.

Вычисление

1. Необходимо сгенерировать случайное k, $1 < k < q$.
2. $R = (\alpha^k \bmod p) \bmod q$ – первая часть подписи.
3. $H = Hash(m)$, где $Hash$ – хеш-функция, описанная в стандарте ГОСТ Р 34.11-94, m – подписываемое сообщение.
4. $S = kH + zR \bmod q$, где z – закрытый ключ.
5. Если $S=0$, то повторить операции 1–4.

Проверка

1. $0 < R < q$ или $0 < S < q$. Если хотя бы одно из двух условий не выполнено, то подпись недействительна. Иначе:
2. $R' = (\alpha^{R/H} y^{S/H} \bmod p) \bmod q$, где y – открытый ключ.
3. Если $R = R'$, то подпись действительна.

4.6 СЛЕПАЯ ПОДПИСЬ НА ОСНОВЕ СТАНДАРТА СТБ 1176.2-99

Стандарт Республики Беларусь предусматривает следующий протокол генерации слепой подписи к документу M:

1. Необходимо сгенерировать случайное k такое, что $1 < k < q$ и вычислить $R = a^k$. Эти действия выполняет подписывающий. Далее он передаёт число R пользователю.
2. Пользователь генерирует случайные числа ε и τ такие, что $1 < \varepsilon, \tau < q$ и затем вычисляет $R' = R * y^{\tau} * a^{\varepsilon}$, $E' = F_H(R' \,||\, M)$ и $E = E' - \tau \pmod q$; E – первый элемент подписи – направляется подписывающему.
3. Подписывающий вычисляет второй элемент подписи $S = (k - xE) \bmod q$ и передаёт S пользователю.
4. Пользователь вычисляет $S' = S + \varepsilon \pmod q$.

В данном описании использованы следующие параметры: q – модуль, по которому ведутся вычисления, простое; a – порождающий элемент; x – закрытый ключ; y – открытый ключ.

4.7 КОЛЛЕКТИВНАЯ СЛЕПАЯ ПОДПИСЬ

Пусть $y_1,\ldots,y_s$ – открытые ключи, которыми владеют s пользователей. Пусть есть сообщение M, которое хотят подписать m из них. В таком случае все подписи можно объединить в одну, длина которой равна длине подписи одного пользователя и не зависит от m. Это реализуется по правилам следующего протокола:

1. Каждый из m пользователей генерирует случайное число $t_{\alpha j} < p$, где $j = 1,\ldots,m$, p – большое простое число. Далее каждый из m пользователей вычисляет

$R_{aj} = (t_{aj})^k \bmod p$ (k - большая простая степень) и рассылает это число всем остальным пользователям из данной группы.

2. Каждый пользователь вычисляет $R = \prod_{j=1}^{m} R_{aj} \bmod p$. Далее вычисляется

$e = f(R,M) = RH \bmod q$, где q – большое простое число, отличное от p,

$H = Hash(M)$ – хеш-функция. Число e будет первым элементом коллективной подписи.

3. $S_{aj} = x^e_{aj} t_{aj} \bmod p$ – доля пользователя. Эту долю каждый пользователь вычисляет и предоставляет остальным.

4. Каждый из пользователей далее вычисляет $S = \prod_{j=1}^{m} S_{aj} \bmod p$. Это второй элемент коллективной подписи.

Проверка коллективной слепой подписи

$y = \prod_{j=1}^{m} y_{aj} \bmod p$ – коллективный открытый ключ. На его основе происходит проверка коллективной слепой подписи по следующему алгоритму:

1. Вычисляется $R = S^k y^{-e} \bmod p$.
2. Вычисляется $e' = f(R,M) = RH \bmod q$.
3. Если $e' = e$, то подпись действительна.

5 ПРОТОКОЛЫ ОБМЕНА КЛЮЧЕВОЙ ИНФОРМАЦИЕЙ
5.1 ПРОТОКОЛ ДИФФИ-ХЕЛЛМАНА

По аналогии между экспоненцированием элементов в мультипликативной группе поля и k-кратным сложением точки эллиптической кривой можно построить протокол Диффи-Хеллмана.

Обмен ключами с использованием эллиптических кривых выполняется по следующей схеме. Сначала выбирается простое число $p \approx 2^{180}$ и параметры a и b для эллиптической кривой в уравнении $y^2 = x^3 + ax + b (\mathrm{mod} p)$. Это задает группу точек на эллиптической кривой. Затем в этой группе выбирается генерирующая точка $G = (x_1, y_1)$. При выборе G важно, чтобы наименьшее значение n, при котором $nG = O$, оказалось очень большим простым числом. Параметры p, a, b и G криптосистемы являются параметрами, известными всем участникам.

Обмен ключами между пользователями A и B проводится:

1. Сторона A выбирает целое число n_A, меньшее n. Это число будет личным ключом участника A. Затем участник A генерирует открытый ключ $P_A = n_A G$.

Открытый ключ представляет собой некоторую точку из группы точек на эллиптической кривой.

2. Точно так же участник B выбирает личный ключ n_B, и вычисляет открытый ключ P_B.

3. Участник A генерирует секретный ключ $K = n_A P_B$, а участник B генерирует секретный ключ $K = n_B P_A$.

Два последних выражения дают один и тот же результат, поскольку

$n_A P_B = n_A (n_B G) = n_B (n_A G) = n_B P_A$.

Чтобы взломать эту схему, противник должен будет вычислить k по данным *G* и *kG*, что предполагается трудной задачей.

Общий секретный ключ представляет собой пару чисел. Если этот ключ предполагается использовать в качестве сеансового ключа для симметричного шифрования, то из этой пары чисел необходимо генерировать одно подходящее значение, например, использовать просто координату *x* или некоторую простую функцию от *x*.

Пример 4. Возьмем $p = 211$, $E_p(0,-4)$ (что соответствует кривой $y^2 = x^3 - 4$) и $B = (2, 2)$. Можно подсчитать, что $241B = O$. Личным ключом пользователя А является $a = 121$, поэтому открытым ключом *A* будет $aB = 121(2, 2) = (115, 48)$. Личным ключом пользователя Б является $b = 203$, поэтому его открытым ключом будет $203(2, 2) = (130, 203)$. Общим секретным ключом является $121(130, 203) = 203(115, 48) = (161,169)$.

Протокол Диффи-Хеллмана, однако, не защищен от противника С, который имеет доступ к каналу связи и может подменять пересылаемые точки P_A и P_B своими точками $P_C = n_C G$. Он, таким образом, может либо выступать от имени одного из пользователей, установив секретную связь с другим, либо, контролируя канал, быть транслятором их переписки, свободно расшифровывая и читая все сообщения. Такого активного криптоаналитика С называют «man in the middle». Для защиты от перехвата и подлога чрезвычайно важной становится задача аутентификации пользователей. Рассмотрим протокол, в котором устраняется недостаток протокола Диффи-Хеллмана.

5.2 ПРОТОКОЛ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ MQV

Протокол распределения ключей на эллиптической кривой (ECKEP-Elliptic Curve Key Establishment Protocol), предложенный Менезисом, Кью и Ванстоуном (A.Menezes, M. Qu, S. Vanstone) и называемый *MQV* - протоколом, предполагает использование как долговременных ключей d_A и d_B, так и разовых ключей k_A, k_B пользователей. Для формирования общего секретного ключа KAB пользователи выполняют следующие действия:

1. Пользователь А:

а) генерирует случайное целое число $0 < k_A < n$;

б) вычисляет точку $R_A = k_A G = (x_A, y_A)$;

в) вычисляет точку $Y_A = k_A Q_B = (x_1, y_1)$;

г) вычисляет целое число $S_A = (k_A + d_A x_A x_1) \bmod n$;

д) отправляет точку R_A пользователю В.

2. Пользователь В:

а) генерирует случайное целое число $0 < k_B < n$;

б) вычисляет точку $R_B = K_B G = (x_B, y_B)$;

в) вычисляет точку $Y_B = k_B Q_A = (x_2, y_2)$;

г) вычисляет целое число $S_B = (k_B + d_B x_B x_2) \bmod n$;

д) отправляет точку R_B пользователю А.

3. Пользователь А:

а) вычисляет точку $Y_B = d_A R_B = (x_2, y_2)$;

б) вычисляет точку $K = S_A(R_B + x_B x_2 Q_B)$.

4. Пользователь B:

а) вычисляет точку $Y_A = d_B R_A = (x_1, y_1)$;

б) вычисляет точку $K = S_B(R_A + x_A x_1 Q_A)$.

В результате вычислений пользователи получают одну и ту же точку *K*.

Действительно, для пользователя A с учетом равенства $Q_B = d_B G$ и имеем

$K = S_A(R_B + x_B x_2 Q_B) = S_A(k_B + d_B x_B x_2)G$.

Аналогично вычисления пользователя B дают

$K = S_B(R_A + x_A x_1 Q_A) = S_B(k_A + d_A x_A x_1)G$.

В связи с коммутативностью операции сложения в группе точек E результаты совпадают. Координаты секретной точки K могут быть известным способом преобразованы в разовый ключ симметричного шифрования.

Протокол *MQV* можно модифицировать без расчета параметров S_A, S, а вычисляя точку *K* тождественными операциями с точками кривой E. Для этого в пункте 3 пользователь A вычисляет точку $V = (R_B + x_B x_2 Q_B)$, а затем точки $k_A V$ и $d_A x_A x_1 V$. Сумма двух последних точек дает точку K. Так же действует и пользователь B. Этот алгоритм более трудоемок, так как операции с точками сложней операций в поле F_q.

При выполнении этого протокола у активного криптоаналитика C («man in the middle») возникают проблемы. Если бы пользователи просто складывали свои секретные ключи, т. е. $S_{A,B} = (k_{A,B} + d_{A,B}) \bmod n$, то ситуация для C будет такой же благоприятной, как и в протоколе Диффи-Хеллмана. Идея защиты от навязывания противником C своего разового ключа k_C состоит в том, что соотношение

$S_A = (k_A + d_A x_A x_1) \bmod n$

нелинейно связывает ключи k_A, d_A и d_B, так как $x_1 = f(k_A, k_B)$. Тем самым противник C лишается возможности свободной подтасовки ключа k_c. Чтобы рассчитать свое значение S_A, в котором ему известно лишь x_A, противнику C придется определить долговременные ключи d_A и d_B, т. е. дважды вычислить дискретный логарифм в группе E.

6 ПРОТОКОЛЫ ТАЙНОГО ГОЛОСОВАНИЯ

В криптографии протоколы тайного голосования – протоколы обмена данными для реализации безопасного тайного электронного голосования через Интернет при помощи компьютеров, телефонов или других специальных вычислительных машин. Это направление криптографии всё ещё развивается, но уже применяется на практике.

Требования к системам тайного голосования

Обязательные:

– никто, кроме голосующего, не должен знать его выбор;

– только легитимные участники могут проголосовать, и притом только один раз;

– решение голосующего не может быть тайно или явно кем-либо изменено (кроме, возможно, как им самим).

Желательные:

– каждый легитимный участник может проверить, правильно ли зачтён его голос;

– каждый легитимный участник может передумать и изменить свой выбор в течение определённого периода времени;

– система должна быть защищена от продажи голосов избирателями;

– в случае, если голос зачтён неправильно, каждый легитимный участник может сообщить об этом системе, не раскрывая своей анонимности;

– невозможно отследить, откуда дистанционно проголосовал избиратель;

– аутентификация оператора;

– можно узнать, кто принимал участие в голосовании, а кто – нет;

– поддержание системы не должно требовать много ресурсов;

– система должна быть отказоустойчива в случае технических неисправностей (потеря электропитания), непреднамеренных (потеря избирателем ключа) и злоумышленных (намеренная выдача себя за другого избирателя, DoS/DDoS) атак.

6.1 ПРОСТОЙ ПРОТОКОЛ ТАЙНОГО ЦИФРОВОГО ГОЛОСОВАНИЯ

Простой алгоритм электронного голосования представляет собой переписку с электронными подписями между избирательным комитетом и множеством избирателей. Пусть здесь и далее: A – агентство, проводящее электронное голосование (англ. agency), E – избиратель, легитимный участник голосования (англ. elector), B – цифровой бюллетень. B может содержать число, имя кандидата, развёрнутый текст или какие-либо другие данные, сообщающие о выборе E, верифицирующие его или необходимые для усиления безопасности протокола. Ход голосования выглядит так:

Алгоритм

Шаг 1. A выкладывает списки возможных избирателей.

Шаг 2. Пользователи, в числе которых и E, сообщают о желании участвовать в голосовании.

Шаг 3. A выкладывает списки легитимных избирателей.

Шаги 1–3 обязательны. Основная цель – определение и объявление числа активных участников n. Хотя некоторые из них могут не участвовать, а некоторые – и вовсе не существовать («мёртвые души», злонамеренно внесённые A), возможность манипулирования голосованием у A заметно снижена. В дальнейшем эти шаги будут считаться за один шаг «утвердить списки».

Шаг 4. A создаёт открытый и закрытый ключ a_{public} и $a_{private}$ и выкладывает в общий доступ a_{public}. Кто угодно может зашифровать сообщение при помощи a_{public}, но расшифровать его сможет только A.

Шаг 5. E

– создаёт собственные публичный и приватный ключи ЭЦП e_{public} и $e_{private}$, затем публикует открытый ключ. Кто угодно может проверить документ E, но подписать его – только сам избиратель. Этот шаг пропускается, если A уже знает электронные подписи избирателей (например, они были сгенерированы при регистрации в системе);

– формирует сообщение B, где тем или иным способом выражает свою волю;

– подписывает сообщение личным закрытым ключом $e_{private}$;

– шифрует сообщение открытым ключом a_{public} ;

– отправляет шифрованное сообщение A;

Шаг 6. A

– собирает сообщения;

– расшифровывает их при помощи лежащего в открытом доступе e_{public} ;

– подсчитывает их и публикует результаты.

Особенности, преимущества и недостатки

Этот протокол чрезвычайно прост, тем не менее, его достаточно, чтобы защититься от внешнего вмешательства, подделки голосов и дискредитации легитимных избирателей. Однако голосующим приходится абсолютно доверять A, ведь его работа никем не контролируется. С одной стороны, E может предоставить злоумышленнику-покупателю голосов доказательство, как он проголосовал, а с другой – не может проверить, что A правильно учёл или даже получил его бюллетень. Поэтому метод применим только в сообществах, где все доверяют друг другу и агентству, отвечающему за подсчёт голосов.

6.2 ПРОТОКОЛ ДВУХ АГЕНТСТВ

Он же протокол Нурми – Саломаа – Сантина. Основная идея состоит в том, чтобы заменить одно избирательное агентство двумя, чтобы они контролировали друг друга. Пусть здесь и далее V – регистратор (англ. validator), в обязанности которого входит подготовка списков, а также допуск или недопуск участника до голосования. Последовательность действий выглядит так:

Алгоритм

Шаг 1. V

– создает набор опознавательных меток t_i и утверждает список возможных избирателей;

– отправляет по защищённому каналу по одной метке каждому голосующему;

– отправляет A весь набор меток без информации о том, какая метка кому принадлежит.

Шаг 2. E

– генерирует e_{public}, $e_{private}$ (для цифровой подписи) и e_{secret} (для того, чтобы ни A, ни посторонний злоумышленник не мог до нужного времени узнать содержимое бюллетеня);

– e_{public} публикуется;

– формирует сообщение B с выбранным решением;

– подписывает его $e_{private}$;

– прикладывает к нему полученный ti ;

– шифрует при помощи e_{secret} ;

– снова прикладывает к шифротексту t_i ;

– отправляет шифротекст $\{t_i, encrypt(e_{secret}, \{t_i, sign(e_{private}, B)\})\}$ на рассмотрение в A

Шаг 3. A

– получает шифротекст. По внешнему тегу оно определяет, что сообщение пришло от легитимного пользователя, но не может определить, ни от какого, ни как он проголосовал;

– выкладывает в открытый доступ полученную пару тег-шифр;

Шаг 4. Опубликованный файл служит сигналом E отправить секретный ключ e_{secret}

Шаг 5. A

– собирает ключи;

– расшифровывает сообщения;

– производит подсчёт голосов;

– присоединяет к опубликованному шифротексту бюллетень без опознавательного тега, на чём голосование заканчивается.

Особенности, преимущества и недостатки

Благодаря выкладыванию в общий доступ полученного файла на шаге 3, A не может впоследствии отрицать получение сообщения от E. При помощи пары «шифр – бюллетень» каждый избиратель может проверить, правильно ли был учтён его голос, что устраняет проблему с недостатком контроля над A. Однако такой подход лишь частично решает проблему необходимости абсолютного доверия к агентству. В случае, если A и V удаётся сговориться, A может манипулировать голосованием. Если агентству известно, кто скрывается под каким опознавательным тегом, оно может специально не принимать сообщения от некоторых избирателей. Кроме того, присутствует проблема «мёртвых душ». Если V внесёт в список заведомо несуществующих избирателей, то A сможет фальсифицировать бюллетени от них.

В протоколах с двумя агентами голосующему не обязательно авторизоваться и перед регистратором, и перед избирательным комитетом. Если избиратель докажет свою личность регистратору, тот может поставить подпись на бюллетень или ключ избирателя. Именно она в дальнейшем будет играть роль допуска до голосования. Кроме того, не обязательно использовать именно метки для авторизации пользователя. По этим причинам в дальнейших алгоритмах конкретный способ идентификации пользователя будет опущен.

6.3 ПРОТОКОЛ ФУДЗИОКИ-ОКАМОТО-ОТЫ

Схема Фудзиоки-Окамото-Оты основывается на протоколе двух агентств и криптографической подписи вслепую. Несильно усложняя протокол, эта схема частично решает проблему сговора двух агентств. Для работы протокола необходим заранее выбранный способ маскирующего шифрования, под которым избиратель присылает регистратору бюллетень. Ослепляющее (маскирующее) шифрование – особый вид шифрования, позволяющее удостовериться в том, что документ подлинный и подписан авторизованным пользователем, но не даёт узнать содержащиеся в нём данные. Маскирующее шифрование должно быть коммутативным с электронной подписью, то есть sign(blind(B)) = blind(sign(B)).

Алгоритм

Шаг 1. V утверждает списки легитимных избирателей

Шаг 2. E

– создаёт e_{public}, $e_{private}$ (для цифровой подписи) и e_{secret} (для того, чтобы ни A, ни посторонний злоумышленник не мог до нужного времени узнать содержимое бюллетеня);

– подготавливает сообщение B с выбранным решением;

– шифрует его e_{secret} ;

– накладывает слой ослепляющего шифрования;

– подписывает его $e_{private}$;

– отправляет V blind(sign($e_{private}$, encrypt(e_{secret} ,B))).

Шаг 3. V

– создаёт v_{public} и $v_{private}$, публичный ключ выкладывается в общий доступ;

– удостоверяется, что бюллетень действительный и принадлежит легитимному и не голосовавшему избирателю;

– подписывает его $v_{private}$;

– возвращает его E.

Шаг 4. E снимает с бюллетени слой маскирующего шифрования (в силу коммутативности остаётся sign($v_{private}$, sign($e_{private}$, encrypt(e_{secret} ,B)))) и отправляет её A

Шаг 5. A

– проверяет подписи E и V;

– помещает всё ещё зашифрованную e_{secret} бюллетень в специальный список, который будет опубликован, после того как все избиратели проголосуют или по истечении заранее оговорённого срока.

Шаг 6. После того как список появляется в открытом доступе, E высылает A e_{secret}

Шаг 7. A

– расшифровывает сообщение;

– подсчитывает результаты.

6.4 ПРОТОКОЛ SENSUS

Лорри Кранор и Рон Ситрон (англ. Lorrie Faith Cranor, Ron K. Cytron) предложили модификацию протокола Фудзиоки – Окамото – Оты под названием Sensus. Отличие заключается в шагах 5–6. После того как A получило зашифрованное сообщение от E, оно не только добавляет его в публикуемый список, а вдобавок отправляет подписанный бюллетень обратно избирателю в качестве квитанции. Таким образом, E не нужно ждать, пока проголосуют все остальные, и он может закончить голосование за один сеанс. Это не только удобно для конечного пользователя, но ещё и предоставляет дополнительного доказательство, что E участвовал в выборах. Кроме того, в Sensus регламентированы дополнительные вспомогательные модули, упрощающие и автоматизирующие ход голосования.

Особенности, преимущества и недостатки

Даже если агентствам удастся сговориться, A не сможет опознать избирателей до того, как получит ключ. Хотя оно всё ещё имеет возможность не принимать сообщения, отпадает возможность игнорировать сообщения конкретно от «неугодных» избирателей. Остаётся лишь проблема подачи голосов за избирателей, не пришедших на выборы. Кроме того, чтобы позволить избирателю переголосовать, в том числе и из-за технической ошибки, необходим дополнительный модуль.

На данный момент протокол Фудзиоки-Окамото-Оты (а также его модификации, включая и Sensus) является одним из самых проверенных протоколов дистанционного электронного голосования.

6.5 ПРОТОКОЛ ХЭ – СУ

Ци Хэ и Чжунминь Су (Qi He, Zhongmin Su) представили ещё более продвинутый по сравнению с Sensus протокол голосования. Этот алгоритм удовлетворяет большей части требований к безопасному протоколу цифрового голосования. Как и Sensus, протокол Хэ–Су использует идею слепой подписи, но подписывается не бюллетень избирателя, а его ключ. Это позволяет голосующим изменять своё решение до конца голосования и ещё больше сужает возможности регистратора и избирательного агентства в случае сговора. Для этого протокола требуется заранее оговорённый способ ослепляющего шифрования и хеш-функция h( ). Как и в протоколе Фудзиоки–Окамото–Оты, маскирующее шифрование должно быть коммутативным с электронной подписью V: sign(blind(B)) = blind(sign(B)), а также

sign(AB) = sign(A)sign(B).

Алгоритм

Шаг 1. V

– утверждает списки легитимных избирателей;

– создаёт v_{public} и $v_{private}$ (используются для асимметричного шифрования);

– v_{public} выкладывается в свободный доступ.

Шаг 2. E

– создаёт e_{public} и $e_{private}$ (используются для подписей);

– вычисляет хеш-функцию от публичного ключа: $h(e_{public})$;

– накладывает слой маскирующего шифрования на $h(e_{public})$. Так как шифруется только хеш от ключа, а не длинное сообщение, можно выбрать какой-нибудь простой способ. Например, E может сгенерировать случайное число x и вычислить $f = encrypt(v_{public}, x)h(e_{public})$;

– отправляет f V.

Шаг 3. V

– проверяет легитимность избирателя;

– дешифрует

f: $g = decrypt(v_{private}, encrypt(v_{public}, x)\ h(e_{public})) = xdecrypt(v_{private}, h(e_{public}))$.

Часть $e_{publi}^{signed} = decrypt(v_{private}, h(e_{public}))$ считается подписанным ключом;

– отправляет g E.

Шаг 4. E

– снимает слой ослепляющего шифрования (умножает на обратный элемент x) и получает подписанный ключ e_{public}^{signed} ;

– проверяет подлинность подписи регистратора: выполняется ли

$encrypt(v_{public}, decrypt(v_{private}, h(e_{public}))) = h(e_{public})$;

– отправляет A пару $\{e_{public}, e_{public}^{signed}\}$.

Шаг 5. A

– как и E, проверяет подлинность подписи регистратора;

– проверяет, совпадает ли хеш-функция от e_{public} в паре с той, что хранится в e_{public}^{signed} ;

– добавляет e_{public} в список авторизированных ключей и сообщает об этом E.

Шаг 6. E

– создает e_{secret} (используется для шифровки бюллетеней, чтобы ни A, ни внешний злоумышленник до нужного времени не мог узнать содержимое бюллетеня);

– подготавливает сообщение B с выбранным решением;

– отправляет A набор $\{e_{public}, encrypt(e_{secret}, B), sign(e_{private}, h(encrypt(e_{secret}, B)))\}$.

Шаг 7. A

– проверяет авторизованность ключа;

– проверяет подлинность сообщения сравнивая хеш зашифрованного сообщения и хеш, полученный при помощи $e_{private}$;

– публикует тройку в открытом списке.

Шаг 8. Появление тройки в открытом списке сигнализирует E отправить A новый набор: $\{e_{public}, e_{secret}, sign(e_{private}, h(e_{secret}))\}$.

Шаг 9. A

– проверяет подлинность сообщения, сравнивая хеши;

– расшифровывает ранее полученную бюллетень;

– публикует все данные;

– подсчитывает результат.

Шаг 10. После голосования V публикует список всех зарегистрировавшихся избирателей, а A – список всех авторизованных ключей.

Особенности, преимущества и недостатки

Схема Хэ-Су удовлетворяет почти всем требованиям к протоколу тайного голосования. Остаётся только повышенный стимул купли/продажи голосов. У A и V теперь нет возможности сжульничать, так как теперь публикуются все списки: возможных избирателей, зарегистрировавшихся и авторизированных ключей. Соответственно, нельзя ни внести несуществующих избирателей, ни голосовать за существующих, но не пришедших. При этом во время составления этих списков ни избирательное агентство, ни регистратор дополнительной информации не получают. У избирателей есть возможность изменить голос. Основной недостаток протокола Хэ – Су – его сравнительная сложность. Так как для поддержания протокола необходимо большое количество ресурсов, он уязвим перед DoS-атаками.

6.6 Протокол на основе ANDOS

За основу взят протокол ANDOS (англ. All or Nothing Disclosure Of Secrets). Идея состоит в том, чтобы усилить стойкость протокола за счёт замены заранее выбранного шифрования секретным ключом на хеширование пользовательской функцией. Далее описано ядро алгоритма. В описании для краткости опущены меры предосторожности и безопасности. В случае необходимости можно применить методы криптографии на открытых ключах и электронной подписи. Предполагается, что для защиты от вмешательства кого-либо извне избиратели могут ещё и мешать информацию между собой, но тогда злонамеренный избиратель может вносить помехи в голосование, поэтому этот шаг тоже пропущен.

Алгоритм

Шаг 1. A

– утверждает список участников голосования;

– пусть набралось n легитимных избирателей. Тогда A выбирает не менее чем n идентификаторов и применяет ANDOS протокол к голосующим. Идентификаторы – большие простые числа, они пронумерованы как 1, . . . ,n.

Шаг 2. E

– выбирает номер i и получает i-е простое число p_i из списка (A ничего не знает о взаимосвязи между i и E);

– выбирает криптографическую хеш-функцию $h_E(x,y)$ двух переменных;

– пересылает A пару $(p_i, h_E(p_i, v_E))$, где v_E – выбор (имя кандидата или в более общем виде, выборочная стратегия) выраженный численно.

Шаг 3. A публикует $h_E(p_i, v_E)$**.**

Шаг 4. После появления $h_E(p_i, v_E)$ в открытом списке E отправляет A пару (p_i, h_E^{-1}). Считая, что y всегда может быть получен по заданным $h_E(x,y)$,x и h_E^{-1}, A теперь знает связь между p_i и v_E (но не между E и его выбором v_E).

Упрощённая версия шагов 2–4 может состоять в том, что E посылает A напрямую пару p_i , v_E. Однако в этом случае будет невозможно для E как проверить засчитан ли голос правильно, так и переголосовать на более поздней стадии. Так может выйти, так как если A публикует идентификатор p_i в списке тех, кто придерживался стратегии v_E, то E будет точно знать, что его голос зачтён правильно, однако впоследствии кто-нибудь сможет замаскироваться под обладающего идентификатором p_i и изменить голос на угодный ему. С другой стороны, если A публикует только количество участников, придерживающихся определённой стратегии v_E, то участники не могут ничего проверить и A может публиковать любые результаты выборов. Хеш-функции используются для того, чтобы злоумышленники не могли определить количество голосов с определённой стратегией v_E (эта информация оказывается полезной), так как задача нахождения исходных значений вычислительно тяжёлая с учётом характерного времени проведения голосования.

Шаг 5. Когда голосование подходит к концу, A объявляет промежуточные результаты, публикуя списки стратегий (кандидатов) с числами $h_E(p_i, v_E)$, соответствующими участникам, голосовавшим за $v = v_E$.

Шаг 6. Если участник E замечает, что его голос размещён в неверном списке, то он посылает A жалобу в виде тройки $(p_i, h_E(p_i, v_E), h_E^{-1})$, которая явным образом показывает верность либо ошибочность результата.

Спустя некоторое время можно начать процедуру изменения голосов (см. финальный шаг). Более простой вариант (Шаг 7) может быть использован для проведения единственного круга повторного голосования.

Шаг 7. Участник E, который хочет изменить свой выбор, отправляет A тройку $(p_i , h_E(p_i, v_E) , v_E')$, где v_E' – новая стратегия. Когда подходит конец тура изменения голосов, A публикует изменённые результаты. Затем повторяется проверка правильности.

Шаг 7'. Всё как и в шаге 7, но теперь участник E отправляет пару $(p_i , h_E'(p_i , v_E'))$, где h_E' – новая хеш-функция, выбранная E. A удостоверяет получение сообщения, публикуя $h_E'(p_i, v_E')$, после чего E отправляет A пару $(p_i , (h_E')^{-1})$. Теперь A знает взаимосвязь между p_i и v_E'. При повторном подведении результатов $h_E(p_i, v_E)$ удаляется из соответ-

ствующего списка, а $h_E'(p_i , v_E')$ добавляется в список с $v' = v_E'$. Участник E может оспорить результат как и прежде.

По сравнению с шагом 7, шаг 7' имеет преимущество в том, что участники, отличные от E, могут лишь наблюдать, что что-либо исчезло из списка v, но не будут знать, что оно переместилось в список v'.

Особенности, преимущества и недостатки

В протоколе ANDOS возможна ситуация, когда два голосующих выберут одно и то же i, таким образом получив одинаковый идентификатор p_i. Возможные решения этой проблемы:

– A удостоверяется, что число идентификационных номеров настолько больше числа голосующих, что вероятность коллизии пренебрежимо мала.

– При повторном получении идентификатора p_i A публикует пару $(p_i' , h_E(p_i , v_E))$, где p_i' – идентификатор, который был не «на продажу» в протоколе ANDOS. По второй компоненте участник E видит, что он под вопросом и отправляет A пару $(p_i' , h_E(p_i' , v_E))$. Другие участники не могут этого сделать, так как они не знают хеш-функции h_E. Далее процесс идёт по той же схеме: A публикует $h_E(p_i' , v_E)$ и т. д. После получения h_E^{-1} A удостоверяется, что откликнулся нужный участник.

Протокол ANDOS довольно затратен, зато для него не нужен независимый регистратор V. Избирателям необходимо выбирать и пересылать не только идентификаторы, но и хеширующие функции, что может быть сложно или долго. A всё ещё может жульничать, распределяя по своему выбору голоса тех, кто заявил о своём намерении принять участие в голосовании, но так и не совершил свой выбор, а E имеют повышенный стимул купли/продажи голосов, так как можно убедиться в результате сделки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Введение в алгебраические системы. Группы : методические указания к изучению курса "Дискретная математика" для студентов специальностей: 1-53 01 02 "Автоматизированные системы обработки информации", 1-40 03 01 "Искусственный интеллект" и 1-40 01 01 "Программное обеспечение информационных технологий" / Министерство образования Республики Беларусь, Брестский государственный технический университет, Кафедра "Интеллектуальные информационные технологии" ; сост. Т. А. Глущенко, М. В. Хацкевич, А. А. Кот. – Брест : БрГТУ, 2019. – 19 с.

2 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления : ГОСТ 7.1-2003 [Электронный ресурс]. – Режим доступа : http://www.internet-law.ru/gosts/gost/1560/. – Дата доступа : 20.11.2018.

3 Применение эллиптических кривых в криптографии : методические указания к выполнению лабораторных работ по дисциплине "Криптографические методы защиты информации" для студентов специальности 1–40 03 01 "Искусственный интеллект" / Министерство образования Республики Беларусь, Брестский государственный технический университет, Кафедра интеллектуальных информационных технологий ; сост. М. В. Хацкевич. – Брест : БрГТУ, 2012. – 28 с.

4 Теоретико-числовые алгоритмы в криптографии. Методические указания к выполнению лабораторных работ по дисциплине "Криптографические методы защиты информации" для студентов специальности 1-40 03 01 "Искусственный интеллект" / Министерство образования Республики Беларусь, Брестский государственный технический университет, Кафедра интеллектуальных информационных технологий ; сост. М. В. Хацкевич. – Брест : БрГТУ, 2012. – 23 с.

5 Теория сравнений. Методические указания к выполнению лабораторных работ по дисциплине "Криптографические методы защиты информации" для студентов специальности 1-40 03 01 "Искусственный интеллект" / Министерство образования Республики Беларусь, Брестский государственный технический университет, Кафедра "Интеллектуальные информационные технологии" ; сост. М. В. Хацкевич. – Брест : БрГТУ, 2019. – 21 с.

Учебное издание

Составители:
Хацкевич Мария Викторовна
Глущенко Татьяна Александровна
Соловчук Александр Михайлович
Хацкевич Анна Сергеевна

Методические указания к выполнению лабораторных работ по дисциплине

«Криптографические методы защиты информации»
для студентов специальности:
6-05-0611-03 «Искусственный интеллект»

Ответственный за выпуск: Хацкевич М. В.
Редактор: Митлошук М. А.
Компьютерная вёрстка: Соколюк А. П.
Корректор: Дударук С. А.

Подписано в печать 28.12.2023 г. Формат 60x84 $^{1}/_{16}$. Бумага «Performer».
Гарнитура «Arial Narrow». Усл. печ. л. 2,33. Уч. изд. л. 2,5. Заказ № 1418. Тираж 21 экз.
Отпечатано на ризографе учреждения образования «Брестский государственный технический университет». 224017, г. Брест, ул. Московская, 267.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/235 от 24.03.2014 г.