Лекция
Протокол разделения секрета и другие протоколы

Протокол «разделения секрета (данных)»

Разделение секрета необходимо для исключения возможности несанкционированного использования этих данных одной персоной.

В качестве секретных данных могут рассматриваться, например, команды по применению ядерного оружия, либо ключи к зашифрованным файлам, содержащим важную информацию по рецептуре продуктов различного рода, «ноу хау» промышленных процессов и т. п.

Разделение данных предполагает, что для выделения основного секрета необходимо объединение частных секретов (*теней*) нескольких лиц, причем в количестве не менее некоторой заданной величины.

Очевидно, что тайный сговор или ошибочное решение достаточно большой группы лиц, владеющих частными секретами для получения основного секрета, значительно менее вероятно, чем неправомерные действия одного лица. Это и обусловливает необходимость создания протокола разделения секретов.

Простейшая схема разделения секретов – «*Все или никто*»,

В ней только объединение частных секретов *всех* их обладателей позволит им вычислить основной секрет.

Секрет = k.

Схема просто реализуется при помощи чисто случайного генерирования двоичных цепочек секретов $k_i, i=1, 2, \ldots, n-1$, всех пользователей, кроме одного, который получает частный секрет следующего вида:

$$k_n = \oplus \sum_{i=1}^{n-1} k_i \oplus k$$

где *k* – основной секрет, а $\oplus$ означает побитовое суммирование по mod 2.

Восстановление секрета

$$k = k_1 \oplus k_2 \oplus \ldots \oplus k_{n-1} \oplus (\sum_{i=1}^{n-1} k_i \oplus k)$$

Однако в этом случае отсутствие (или потеря) даже одного из частных секретов приведет к невозможности восстановления основного секрета.

Пороговой схема

для восстановления основного секрета достаточно объединить только *m* из *n* частных секретов.

Определение 1. *Пороговой (n, m)-схемой* называется такой алгоритм формирования частных секретов $k_i, i=1, 2, ..., n$ (теней), по основному секрету *k*, что при объединении *m* или более таких теней существует алгоритм, позволяющий восстановить в точности основной секрет, тогда как объединение менее чем *m* теней не дает абсолютно никакой информации об основном секрете *k* (рис. 1).

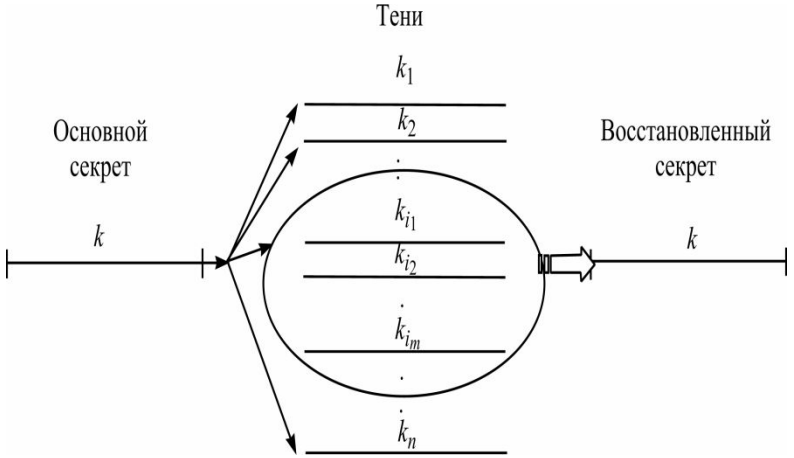


Рис. 1. Пороговая (*n*, *m*)-схема

Существует множество различных способов построения пороговых схем.

Схема разделения секрета на основе интерполяции полиномов над конечными полями (схема Шамира)

Пусть основной секрет $k \in GF(p)$, где $GF(p)$ – конечное простое поле, что всегда возможно для любого *k* при выборе необходимой величины *p*.
Пусть

$$h(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0$$

$m-1$

полином степени m-1, $a_1, \ldots, a_{m-1} \in GF(p)$.

Выберем коэффициент a_0 полинома при его постоянном члене равным основному секрету *k*, а остальные его коэффициенты $a_1, \ldots, a_{m-1} \in GF(p)$ выберем чисто случайными.
Полином (m-1)-й степени однозначно определяется всеми этими коэффициентами

Частные секреты (тени) вычисляются по формуле: $k_i = h(x_i),\ i = 1, 2, \ldots, n,\quad x_i \in GF(p)$ где, в частности, можно взять , $x_i = i$, $n < p$.

Затем k_i передаются каждому из *n* пользователей по секретным каналам. Если *m* (или более) пользователей $i_1, i_2, ..., i_m$ объединят свои индивидуальные секреты $k_{i_1}, k_{i_2}, ..., k_{i_m}$, то они смогут восстановить полином

, используя интерполяционную формулу Лагранжа [14]:

$$h(x) = \sum_{s=1}^{m} k_{i_s} \prod_{\substack{j=1 \\ s \neq j}}^{m} \frac{(x - x_{i_j})}{(x_{i_s} - x_{i_j})} \bmod p$$

Тогда основной секрет легко может быть найден как

$$a_0 = h(0) = \sum_{s=1}^{m} k_s C_s \bmod p \quad , \text{ где} \quad C_s = \prod_{\substack{s'=1 \\ s \neq s'}}^{m} \frac{x_{j_{s'}}}{x_{j_s} - x_{j_{s'}}}$$

j_s - номер пользователя,

Если же число теней оказывается менее *m*, то они не будут содержать никакой информации о коэффициенте a_0, поскольку для любого значения $a_0 \in GF(p)$ всегда найдется полином (*m* –1)-й степени $\tilde{h}(x)$, удовлетворяющий уравнениям $\tilde{h}(x_i)=k_i, i=1, 2, ..., m-1$.

Пример Пусть требуется создать пороговую схему (5, 3) предназначенную для пяти пользователей, в которой для восстановления основного секрета требуется три или более теней.

Тени получаются при помощи вычисления многочлена в *пяти* различных точках. В частном случае первой тенью может быть значение многочлена при $x=1$, второй тенью – значение многочлена при $x=2$ и т. д.

Пусть *k* равно 13. Чтобы создать пороговую (5, 3)-схему, в которой любые три из пяти пользователей смогут восстановить *k*, сначала выберем простое число $p=17$ (оно больше количества теней и больше основного секрета в данном примере).

Предположим, что числа 2 и 10 оказались случайно выбранными коэффициентами полинома, который в этом случае будет равен

$$h(x)=2x^2+10x+13$$

Пятью тенями оказываются тогда следующие числа:

$$k_1=h(1)=(2+10+13)\bmod 17=8$$
$$k_2=h(2)=(8+20+13)\bmod 17=7$$
$$k_3=h(3)=(18+30+13)\bmod 17=10$$
$$k_4=h(4)=(32+40+13)\bmod 17=0$$
$$k_5=h(5)=(50+50+13)\bmod 17=11$$

Эти тени распределяются среди пяти участников протокола. Допустим, 1-й, 3-й и 5-й из них, собрав свои тени, хотят восстановить основной секрет. Используя интерполяционную формулу Лагранжа, они находят

$$h(x)=\left[\frac{8\cdot(x-3)\cdot(x-5)}{(1-3)\cdot(1-5)}+\frac{10\cdot(x-1)\cdot(x-5)}{(3-1)\cdot(3-5)}+\frac{11\cdot(x-1)\cdot(x-3)}{(5-1)\cdot(5-3)}\right]\bmod 17$$

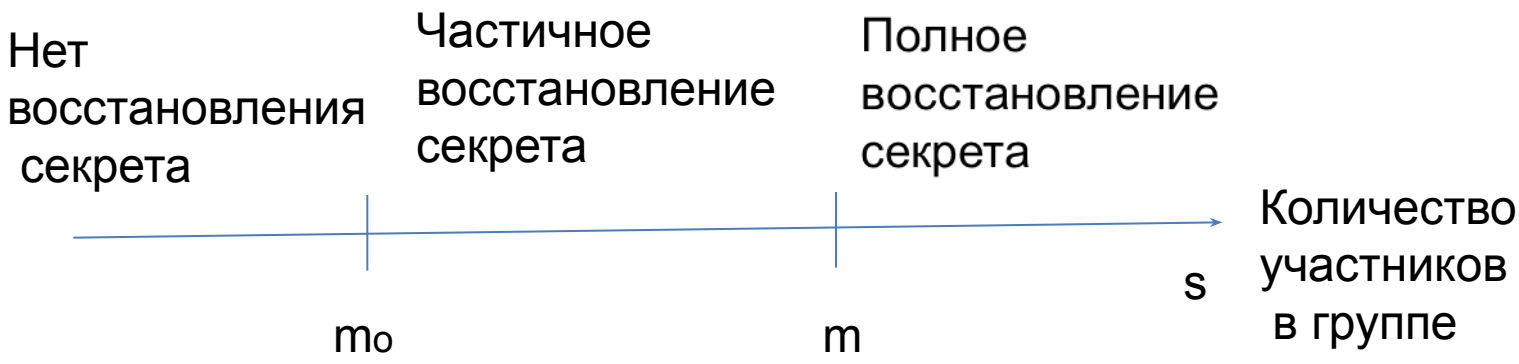
Произведя все вычисления по модулю 17, получаем $h(x)=2x^2+10x+13$.
Свободный член в полученном полиноме 13 и есть восстановленный основной секрет.

Основное преимущество пороговой схемы на основе интерполяционных полиномов Лагранжа состоит в том, что при появлении новых пользователей не надо менять основной секрет, и если, например, он является ключом, на котором зашифрованы некоторые данные, то их не надо перешифровывать, а достаточно лишь вычислить для основного ключа дополнительные индивидуальные секреты.

Имеется множество различных обобщений [14] для пороговых схем разделения секретов, например:

- взвешенные секреты распределения порогов по группам пользователей;
- схемы с обнаружением фальсификации распределяющей стороной (дилером) или отдельными пользователями своих частных данных (теней).

Еще одним интересным обобщением схемы с разделением секретов является так называемая $(n, m, m_0,)$ ***рамп-схема***, в которой объединение *m* и более пользователей однозначно восстанавливает секрет, при объединении теней *s* пользователей в интервале $m_0 \leq s < m$ основной секрет восстанавливается лишь частично, а при числе объединенных пользователей, меньшем, чем m_0 , восстановить его оказывается невозможно.



(n,m)-схема разделения секрета Асмуса-Блума

Основана на использовании простых чисел и Китайской теоремы об остатках.

Пусть M секрет, выберем простое число p>M.

Доверенный центр выбирает n взаимно простых чисел

d1, d2,…,dn, таких что:

- для любого *i* di>p;
- d1<d2….,<dn;
- d1·d2…·dm>p·dn-m+2·dn-m+3……·dn

Затем доверенный центр выбирает случайное число r и вычисляет число M’

M’= M+rp,

Затем для каждого абонента вычисляется число $k_i = M' \bmod d_i$.

Тенью (долью секрета) для i-го абонента будет тройка (p,di,ki)

Восстановление секрета

Пусть m абонентов, хотят вычислить секрет.

Они составляют систему сравнений

M’=k1modd1

M’=k2modd2

…….

M’=kmmoddm

и решают систему относительно M’. Получают секрет

M=M’modp.

Секрет нельзя восстановить для m-1 и менее теней.

Пример схемы Асмуса-Блюма

- Предположим, что нам нужно разделить секрет M = 2 между четырьмя участниками таким образом, чтобы любые три из них могли этот секрет восстановить (а два участника — не могли бы). То есть нужно реализовать (4,3)-пороговую схему
- В качестве простого числа выберем p = 3, в качестве взаимно простых di= {11 , 13 , 17 , 19}
- Проверяем, что:

$\forall\, i : di \in \{ 11 , 13 , 17 , 19 \} , di > p = 3$

$d\,1 < d\,2 < d\,3 < d\,4 \equiv 11 < 13 < 17 < 19$

$d\,1 * d\,2 * \ldots d\,m > p * (d\,n - m + 2) * (d\,n - m + 3) * \cdots * d\,n$

$d\,1 * d\,2 * d\,3 > p * d\,3 * d\,4 \qquad 11 * 13 * 17 > 3 * 17 * 19$

- Выбираем случайное число r = 51 и вычисляем:

$M' = M + r\,p = 2 + 51 * 3 = 155$

- Вычисляем доли:

$k_i = M' \bmod d_i$

$k_1 = 155 \bmod 11 = 1$

$k_2 = 155 \bmod 13 = 12$

$k_3 = 155 \bmod 17 = 2$

$k_4 = 155 \bmod 19 = 3$

Теперь попробуем восстановить исходный секрет, имея на руках доли (ki, kj, ks,): { 1 , 12 , 2 }, { 12 , 2 , 3 } , { 2 , 3, 1} , {3,1,12}

- Составим систему уравнений для (k1, k2, k3) :

$1 = M' \bmod 11$

$12 = M' \bmod 13$

$2 = M' \bmod 17$

- Восстанавливаем M ′=155, решая систему основываясь на китайской теореме об остатках.
- Зная M ′ , мы вычисляем секрет M.

$M \equiv M' \bmod p \qquad M \equiv 155 \bmod 3 \equiv 2$

Замечание:

Так как в данном примере 155<17*19 , то два участника восстановят секрет. Поэтому M' должно быть больше произведения долей неавторизованных участников !

Проверяемое разделение секрета

Задача разделения секрета возникает тогда, когда по разным причинам владелец секрета (дилер) не полностью доверяет участникам.
Поэтому следует ожидать, что участники также не доверяют дилеру.
По этой причине на практике необходимы схемы **проверяемого разделения секрета.**

Схемы проверяемого разделения секрета представляют собой класс схем разделения секрета с защитой от нечестного дилера. Они предоставляют всем участникам разделения возможность проверить, что ими были получены корректные доли секрета (т.е. доли воссоздают одинаковый секрет). Другими словами, такая схема гарантирует существование секрета, который участники смогут восстановить в дальнейшем.

Протокол проверяемого разделения секрета Фельдмана

В основе создания неинтерактивного протокола проверяемого разделения секрета Фельдмана лежит идея сочетание схемы разделения секрета Шамира со схемой гомоморфного шифрования. В таком случае для проверки достоверности долей каждого участника используются гомоморфные отношения, которые могут существовать между проверочными значениями и зашифрованными долями.

Выбор параметров схемы: Дилер D, n –участников: $P_1, P_2, ..., P_n$,
Пусть p, q – большие простые числа, $p - 1 \equiv 0 \pmod q$.
g – элемент порядка q группы Z_p^*, т.е. $g^q \equiv 1 \pmod p$
Секрет s

Распределение секрета и проверочных значений

Дилер выбирает многочлен $Q(x) \in Z_q[x]$ с коэффициентами $Q_0 = s, Q_1, \dots, Q_{k-1}$ и раздает всем участникам соответствующие проверочные значения $g^{Q_0}, g^{Q_1}, \dots, g^{Q_{k-1}}$.
Положим $x_i = i, i = \overline{1, n}$.
Дилер секретно передает каждому участнику схемы P_i предназначенную ему долю $s_i = Q(i) \bmod q$

Проверка долей секрета участниками:

Проверочное уравнение для участника P_i:

$$g^{s_i} =^{?} \left(g^{Q_0}\right) * \left(g^{Q_1}\right)^{i} * \left(g^{Q_2}\right)^{i^2} * \cdots * \left(g^{Q_{k-1}}\right)^{i^{k-1}} \pmod{p}.$$

При положительном результате проверки будет выполнено равенство:

$$s_i = Q_0 + Q_1 i + Q_2 i^2 + \cdots + Q_{k-1} i^{k-1} \pmod{q}.$$

Восстановление секрета

Если k (или более) пользователей $i_1, i_2, \ldots, i_k$ объединяют свои индивидуальные ключи $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$, то они смогут восстановить полином $h(x)$, используя интерполяционную формулу Лагранжа:

$$h(x) = \sum_{m=1}^{k} s_{i_m} \prod_{\substack{j=1 \\ j \neq m}}^{k} \frac{(x - x_{i_j})}{(x_{i_m} - x_{i_j})} \; mod\; p, \qquad a_0 = h(0)$$

Пример протокола Фельдмана

Параметры: 5 участников.

Дилером выбраны $p = 2111,\ q = 211$, поскольку данные значения удовлетворяют условию $p - 1 \equiv 0\ (mod\ q)$.

g – элемент порядка q группы Z_p^*, т.е. $g^q \equiv 1\ (mod\ p)$.

$\mathrm{g}^{211} \equiv 1\ (\mathrm{mod}\ 2111)$ => $\mathrm{g} = 3$ (методом перебора для $\mathrm{g} > 1$).

Пусть секрет $s = 15\ (mod\ q)$.

Разделение секрета и вычисление проверочных значений:

Дилер выбирает многочлен $\mathrm{Q(x)} \in \mathrm{Z_q[x]}$ с коэффициентами $\mathrm{Q_0} = \mathrm{s},\ \mathrm{Q_1}, \dots, \mathrm{Q}_{k-1}$:

$Q(x) = Q_2 x^2 + Q_1 x + s = 6x^2 + 9x + 15\ (mod\ 211)$.

Дилер вычисляет проверочные значения, которые раздаются всем участникам

$g^{Q_0}\ (mod\ p) = 3^{15} (mod\ 2111) = 440\ (mod\ 2111)$;

$g^{Q_1}\ (mod\ p) = 3^9\ (mod\ 2111) = 684\ (mod\ 2111)$;

$g^{Q_2}\ (mod\ p) = 3^6\ (mod\ 2111) = 729\ (mod\ 2111)$.

Частные ключи (тени) вычисляются по формуле:

$s_i = Q(x_i)\ mod\ q$, где $x_i = i = \{1, 2, 3, 4, 5\}$:

$s_1 = Q(1) = (6 + 9 + 15)\ mod\ 211 = 30$; $\quad s_2 = Q(2) = (24 + 18 + 15)\ mod\ 211 = 57$;

$s_3 = Q(3) = (54 + 27 + 15)\ mod\ 211 = 96$; $\quad s_4 = Q(4) = (96 + 36 + 15)\ mod\ 211 = 147$;

$s_5 = Q(5) = (150 + 45 + 15)\ mod\ 211 = 210$.

Проверка долей:

После фазы распределения долей и до фазы восстановления секрета каждый из участников может проверить достоверность своей доли. Проверочное уравнение имеет вид:

$$g^{s_i} =^{?} \left(g^{Q_0}\right) * \left(g^{Q_1}\right)^{i} * \left(g^{Q_2}\right)^{i^2} * \cdots * \left(g^{Q_{k-1}}\right)^{i^{k-1}} (\mathrm{mod}\ p).$$

Пусть третий участник решил проверить свою долю. Тогда уравнение будет иметь следующий вид:

Номер участника 3

$$3^{96} \equiv \left(3^{15}\right) * \left(3^{9}\right)^{3} * \left(3^{6}\right)^{3^2} \ mod\ 2111;$$

$$3^{96} \equiv 440 * (684)^{3} * (729)^{9}\ mod\ 2111;$$

$$180 \equiv 440 * 681 * 1452\ mod\ 2111;$$

$180 \equiv 180\ mod\ 2111$ => проверка прошла успешно!

После фазы проверки каждый из участников распространяет сообщение о принятии долей, и, если все P_i, $i = \overline{1, n}$ распространили такое сообщение, фаза распределения долей завершилась успешно.

Восстановление секрета:

Восстановим секрет для 1, 2 и 3 участников посредством использования интерполяционной функции Лагранжа:

$$Q(x) = \left[\frac{30(x-2)(x-3)}{(1-2)(1-3)} + \frac{57(x-1)(x-3)}{(2-1)(2-3)} + \frac{96(x-1)(x-2)}{(3-1)(3-2)}\right] mod\ q =$$

$$= \left[\frac{30(x^2-5x+6)}{2} + \frac{57(x^2-4x+3)}{-1} + \frac{96(x^2-3x+2)}{2}\right] mod\ 211 =$$

$$= \left[15\left(x^2-5x+6\right) - 57\left(x^2-4x+3\right) + 48(x^2-3x+2)\right] mod\ 211 =$$

$$= \left[15x^2 - 75x + 90 - 57x^2 + 17x - 171 + 48x^2 - 144x + 96\right] mod\ 211 =$$

$$= 6x^2 + 9x + \mathbf{15}\ mod\ 211.$$

Свободный член в полученном полиноме 15 и есть восстановленный основной секрет.

Стойкость протокола Фельдмана

В отличие от обычных схем разделения секрета, стойкость протокола Фельдмана основывается на предположении о вычислительной сложности задачи дискретного логарифмирования, т.е. нахождения значений s_i при известных значениях $z_i = g^{s_i}$. Следовательно, в то время как обычные схемы разделения секрета требуют, чтобы любое количество участников, не составляющее кворум, не получало никакой информации о секрете, схема проверяемого разделения секрета при наличии такого количества участников также не позволяет восстановить секрет. Однако в рассмотренном примере любой участник мог бы узнать секрет, если бы обладал возможностью вычислять дискретные логарифмы.

Протокол проверяемого разделения секрета Педерсена

Выбор параметров схемы: Дилер, n –участников
Числа p, q, g, s определяются так же, как и в схеме Фельдмана.
$h \in Z_p^*$ – открытое общедоступное число, однако такое, что $g^d = h \pmod p$, где $d \in Z_q$ неизвестно в том числе дилеру.

Разделение секрета
Дилер выбирает два многочлена $F(\cdot)$, $G(\cdot)$ степени $k-1$ над полем Z_q с коэффициентами $F_0 = s$ и случайными коэффициентами $\{F_m\}_{m\in\{1,\dots,k-1\}}$ и $\{G_m\}_{m\in\{0,\dots,k-1\}}$ соответственно, т.е.

$$F(x) = F_0 + F_1 x + F_2 x^2 + \cdots + F_{k-1} x^{k-1} \in Z_q[x],$$
$$G(x) = G_0 + G_1 x + G_2 x^2 + \cdots + G_{k-1} x^{k-1} \in Z_q[x],$$

и распространяет всем участникам схемы P_i, $i = \overline{1,n}$ проверочные значения $E_m = g^{F_m} * h^{G_m} \pmod p$, $m = \overline{0,k-1}$. Затем дилер секретно пересылает всем P_i, $i = \overline{1,n}$ их доли $\{s_i, t_i\}$, где $s_i = F(i)$, $t_i = G(i)$.

Проверка долей секрета

Проверочное уравнение для участника P_i:

$$g^{s_i} h^{t_i} =^{?} (E_0) * (E_1)^i * (E_2)^{i^2} * \cdots * (E_{k-1})^{i^{k-1}} \pmod p.$$

При положительном результате проверки будет выполнено равенство:

$$\left(g^{F_0} h^{G_0}\right) * \left(g^{F_1} h^{G_1}\right)^i * \cdots * \left(g^{F_{k-1}} h^{G_{k-1}}\right)^{i^{k-1}} =$$
$$= g^{F_0 + F_1 i + \cdots + F_{k-1} i^{k-1}} * h^{G_0 + G_1 i + \cdots + G_{k-1} i^{k-1}} =$$
$$= g^{F(i)} * h^{G(i)} \pmod p.$$

Восстановление секрета

Если k (или более) пользователей $i_1, i_2, \ldots, i_k$ объединяют свои индивидуальные ключи $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$, то они смогут восстановить полином $h(x)$, используя интерполяционную формулу Лагранжа:

$$h(x) = \sum_{m=1}^{k} s_{i_m} \prod_{\substack{j=1 \\ j \neq m}}^{k} \frac{(x - x_{i_j})}{(x_{i_m} - x_{i_j})} \; mod\; p, \qquad a_0 = h(0)$$

Пример протокола Педерсена

Инициализация протокола

Пусть дилером выбраны $p = 2111,\ q = 211$, поскольку данные значения удовлетворяют условию $p - 1 \equiv 0\ (mod\ q)$.

g – элемент порядка q группы Z_p^*, т.е. $g^q \equiv 1 (\mathrm{mod}\ p)$.

$g^{211} \equiv 1\ (\mathrm{mod}\ 2111)$ => $g = 3$ (найден методом перебора для $g > 1$).

Пусть секрет $s = 15\ (\mathrm{mod}\ q)$.

$g^d = h\ (\mathrm{mod}\ p), h \in Z_p^*, d \in Z_q$.

$d = 173 => \ h = 3^{173}\ mod\ 2111 = 1920.$

Разделение секрета

Дилер выбирает два многочлена $Q(x)$, $F(x)$ степени $k - 1$ над полем Z_q с коэффициентами $Q_0 = s$ и случайными коэффициентами $\{Q_m\}_{m\in\{1,\dots,k-1\}}$ и $\{F_m\}_{m\in\{0,\dots,k-1\}}$ соответственно:

$Q(x) = Q_2x^2 + Q_1x + s = 6x^2 + 9x + 15\ (mod\ 211)$,

$F(x) = F_2x^2 + F_1x + F_0 = 2x^2 + 10x + 13\ (mod\ 211)$.

Дилер вычисляет проверочные значения, которые распределяют всем участникам разделения:

$E_0 = 3^{15} * 1920^{13}\ mod\ 2111 = 1052;$

$E_1 = 3^9 * 1920^{10}\ mod\ 2111 = 992;$

$E_2 = 3^6 * 1920^2\ mod\ 2111 = 271.$

Частные ключи (тени) вычисляются по формуле
$s_i = Q(x_i)\ mod\ q,\ t_i = F(x_i)\ mod\ q$, где $x_i = i = \{1, 2, 3, 4, 5\}$:
$s_1 = Q(1) = 30,\ t_1 = F(1) = 25 \Rightarrow \{s_1, t_1\} = \{30, 25\};$
$s_2 = Q(2) = 57,\ t_2 = F(2) = 41 \Rightarrow \{s_2, t_2\} = \{57, 41\};$
$s_3 = Q(3) = 96,\ t_3 = F(3) = 61 \Rightarrow \{s_3, t_3\} = \{96, 61\};$
$s_4 = Q(4) = 147,\ t_4 = F(4) = 85 \Rightarrow \{s_4, t_4\} = \{147, 85\};$
$s_5 = Q(5) = 210,\ t_5 = F(5) = 113 \Rightarrow \{s_5, t_5\} = \{210, 113\}.$

Проверка долей

После фазы распределения долей и до фазы восстановления секрета каждый из участников может проверить достоверность своей доли.
Проверочное уравнение имеет вид:

$$g^{s_i} h^{t_i} =^{?} (E_0) * (E_1)^{i} * (E_2)^{i^2} * \cdots * (E_{k-1})^{i^{k-1}} \pmod{p}.$$

Пусть третий участник решил проверить свою долю. Тогда уравнение будет иметь следующий вид:

$3^{96} * 1920^{61} \equiv 1052 * 992^3 * 271^9\ mod\ 2111;$
$638 \equiv 1052 * 1758 * 27\ mod\ 2111;$
$638 \equiv 638\ mod\ 2111$ => проверка прошла успешно!

Восстановление секрета

Восстановим секрет для 1, 2 и 3 участников посредством использования интерполяционной функции Лагранжа:

$$Q(x) = \left[\frac{30(x-3)(x-5)}{(1-3)(1-5)} + \frac{57(x-1)(x-5)}{(3-1)(3-5)} + \frac{96(x-1)(x-3)}{(5-1)(5-3)}\right] mod\ q =$$

$$= \left[\frac{30(x^2-5x+6)}{2} + \frac{57(x^2-4x+3)}{-1} + \frac{96(x^2-3x+2)}{2}\right] mod\ 211 =$$

$$= \left[15\left(x^2-5x+6\right) - 57\left(x^2-4x+3\right) + 48(x^2-3x+2)\right] mod\ 211 =$$

$$= \left[15x^2-75x+90-57x^2+17x-171+48x^2-144x+96\right] mod\ 211 =$$

$$= 6x^2+9x+15\ mod\ 211.$$

Свободный член в полученном полиноме 15 и есть восстановленный основной секрет.

Оценка стойкости

В отличие от первой схемы, здесь, помимо свойства гомоморфизма дискретного логарифма, используется схема обязательства, которая позволяет скрыть секрет, даже если вычислительно неограниченный противник умеет решать задачу дискретного логарифмирования, что обеспечивает теоретико-информационную стойкость протокола.

Одним из свойств рассматриваемой схемы является тот факт, что легко вычислить линейные комбинации общих секретов.

Протокол поручительства информации или Протокол обязательства

Протокол обязательства представляет собой криптографическую схему, которая позволяет зафиксировать некоторое значение (сообщение), сохраняя его скрытым для других, с возможностью раскрытия зафиксированного значения в дальнейшем. Сторона, принявшая обязательство, не может изменить значение после отправки. То есть протокол обязательства реализует связывание данных.

Взаимодействие сторон в схеме обязательств происходит в два этапа:

фаза фиксации или передачи («Commit») – определение фиксируемого значения и передача зашифрованного сообщения от отправителя к получателю (обязательство);

фаза раскрытия («Reveal») – раскрытие фиксируемого значения посредством полученного от отправителя ключа и проверка этого значения.

Поручительство (способ 1)

Поручительство информации

Напомним сущность этого КП. Пользователь сети *A* отдает на хранение (*поручительство)* пользователю *B* некоторую битовую цепочку данных *M* (в частном случае – цепочку длины 1, т. е. один бит). *B* не может прочитать эту цепочку до определенного момента времени, задаваемого *A*, однако и *A* не может изменить позже содержание цепочки *M*, хранящейся у *B*.

Данная задача решается следующим простым протоколом:

1. *B* генерирует случайную битовую цепочку *R* и посылает ее по сети к *A*.

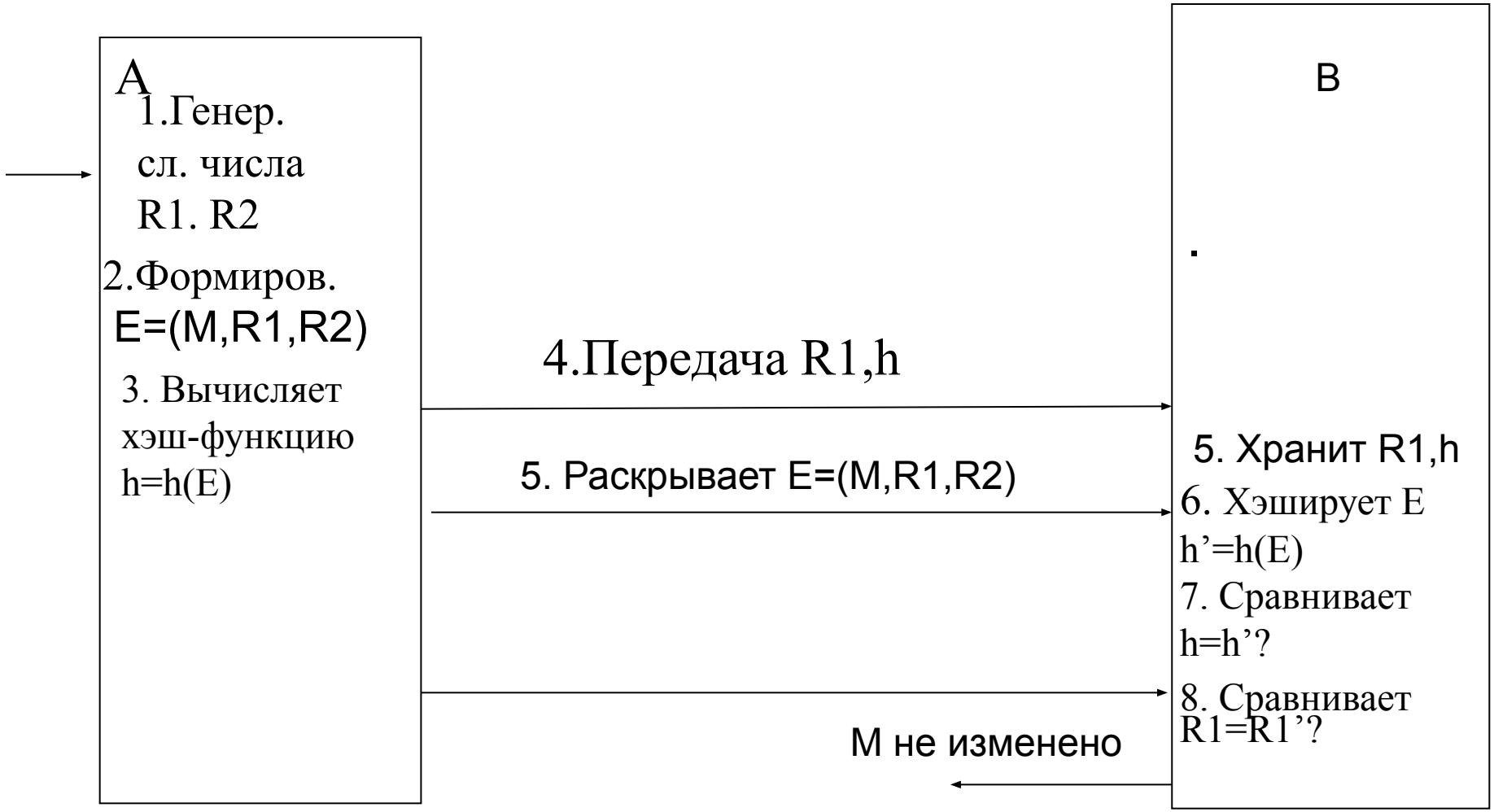
2. *A* объединяет *R* со своим сообщением *M* и посылает *B* криптограмму $E = f_K(R,M)$ $f_K(\cdot)$ где – алгоритм шифрования некоторым стойким блоковым шифром на секретном ключе *K*.

3. *B* хранит *E* до поступления команды от *A*.

4. В определенное время *A* посылает *B* свой секретный ключ *K*, при помощи которого *B* дешифрует *E*.

5. *B* проверяет присутствие своей цепочки *R* в дешифрованном сообщении.
Очевидно, что все задачи протокола решаются после выполнения данных шагов.

Поручительство (способ 2)



Можно использовать для решения той же задачи другой протокол, который не требует выполнения процедур шифрования/дешифрования и активности от пользователя *B*:

1. *A* генерирует случайные числа R_1, R_2 .

2. *A* объединяет их со своим сообщением *M*, формируя цепочку $E = (R_1, R_2, M)$.

3. *A* выполняет хеширование *E* при помощи однонаправленной бесключевой хеш-функции $h(\cdot)$ т. е. вычисляет $h=h(E)$.

4. *A* посылает *B* цепочку h, R_1 .

5. В определенное время *A* посылает *B* цепочку $E = (R_1, R_2, M)$.

6. *B* убеждается в правильности сохраняемого им сообщения *M*, выполняя хеширование цепочки *E*.

Безопасность данного протокола для *B* определяется тем обстоятельством, что *A* не может найти другую такую цепочку (R_1, R'_2, M'), для которой $h(R_1, R'_2, M') = h(R_1, R_2, M)$. (Если бы *A* не посылала *B* цепочку R_1, то он имел бы возможность изменить обе величины R_1, R_2 , а затем подделать сообщение *M*.)

Протокол: доказательство с нулевым разглашением

При помощи протоколов из данного семейства один из пользователей компьютерной сети может доказать другому пользователю, что у него имеется некоторая информация, не раскрывая самой информации.
Доказательства с нулевым разглашением обычно принимают форму интерактивных протоколов.
Участники протокола: ***Проверяющий*** **(V),** ***Доказывающий*** **(P)**

Общая идея всех протоколов:
Проверяющий задает k вопросов доказывающему.
Если P знает секрет, то он ответит на все вопросы V правильно.
Если же секрет ему не известен, то у него есть лишь некоторая вероятность (обычно 0,5) ответить правильно. Тогда после значительного количества вопросов V сможет достоверно убедиться, знает ли P секрет.
Однако ни один из заданных V вопросов и ответов P на них не должен дать V ни малейших сведений об информации, которой обладает P.

Правильно: +1+1+1+…..+1=к
Неправильно: +1-1-1+1….+1-1=0

Простой пример доказательства с нулевым разглашением секрета

Задача: А генерирует число n (n=pq) и передает n В, не раскрывая сомножителей.
В хочет убедиться, что действительно n=pq, но А не может раскрыть p, q . (В не может факторизовать n, так как это вычислительно трудная задача).

Решение: А генерирует множество {N} из S чисел $n_i \quad (n_i = p_i q_i)$ и предлагает В выбрать произвольное подмножество из k этих чисел. В выбирает, после чего А раскрывает множители этих чисел. В убеждается, что действительно любое число состоит из двух множителей. Если предположить, что в множестве {N} для одного из чисел $n \neq p\ q$. То вероятность ошибки равна $(1/S)^k$ При этом А не раскрыл секрет.

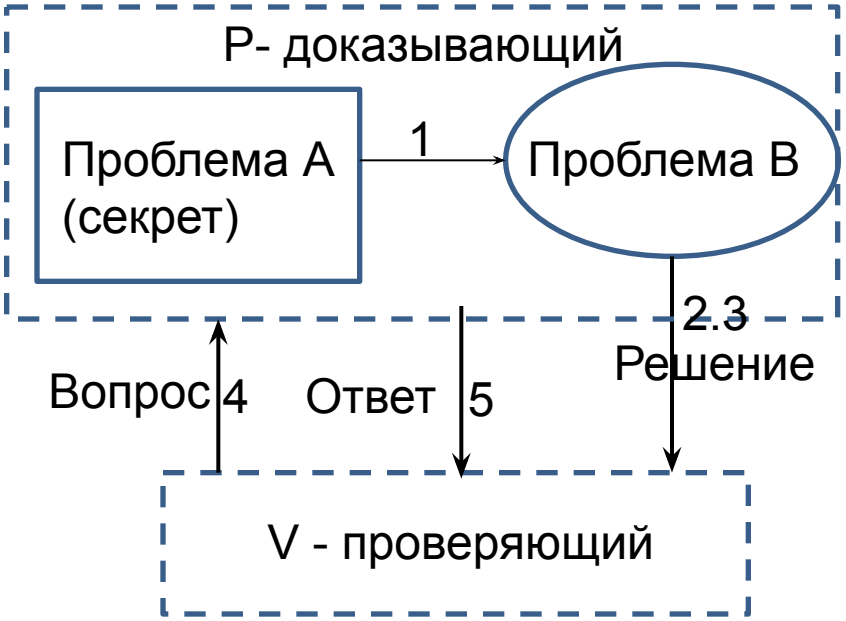
Общая постановка задачи: Доказательство с нулевым разглашением секрета

Предположим, что P известна некоторая информация, которая является решением трудной проблемы. Базовый протокол нулевого разглашения состоит обычно из нескольких обменов :

1. P использует свою информацию и случайное число для преобразования основной трудной проблемы в другую, изоморфную ей проблему. Затем он использует свою информацию и известное ему случайное число для решения новой трудной проблемы.

2. P передает V решение новой проблемы, используя протокол *поручительства информации*.

3. P объясняет V сущность новой трудной проблемы, однако V не может использовать это знание для получения какой-либо информации о первоначальной проблеме или путях ее решения.



4. V просит P одно из двух:

а) доказать ему, что новая и старая проблемы изоморфны,

б) открыть решение, полученное V на этапе шага 2, и доказать, что это действительно решение новой проблемы.

5. P исполняет просьбу V. **6**. P и V повторяют *n* раз шаги 1–5.

Пример трудной задачи

Рассмотрим нахождение так называемого *гамильтонового цикла* на заданном графе.

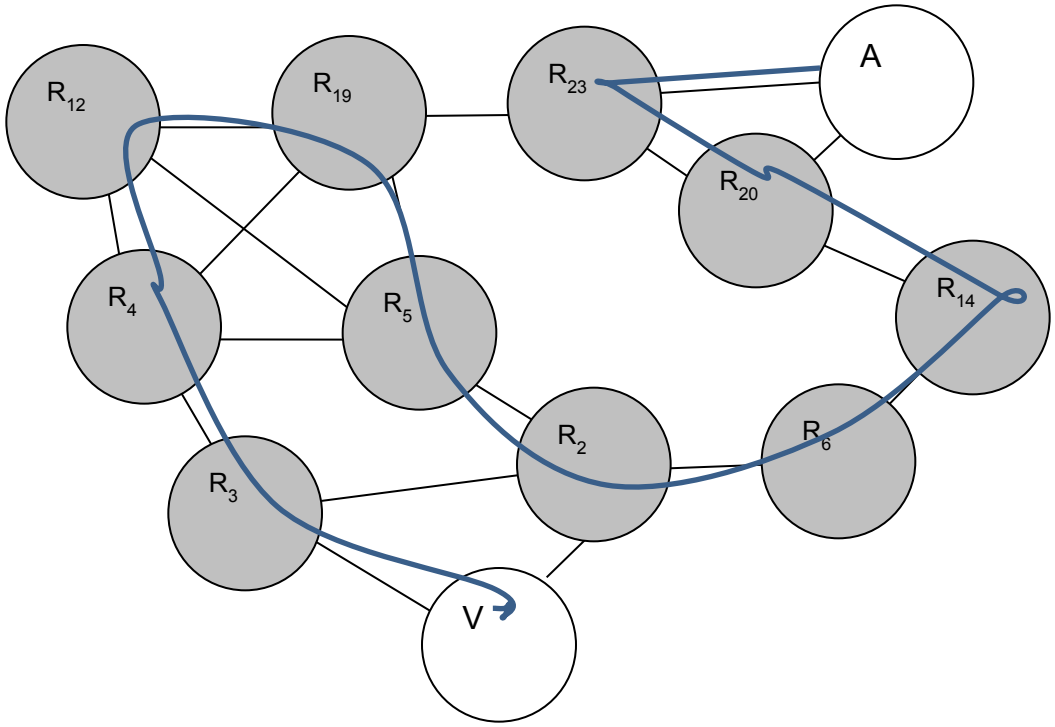
Напомним сначала, что гамильтоновым циклом (ГЦ) называется замкнутая непрерывная линия на графе, которая проходит по ребрам графа через все его вершины только один раз, за исключением исходной вершины.

Не каждый граф содержит ГЦ, и до сих пор не известно общих полиномиально сложных методов установления этого факта, а тем более нахождения самого этого цикла, даже если он существует.

Если два графа идентичны во всем, кроме наименования точек, то они называются *топологически изоморфными*. Для графов очень больших размеров доказательство их изоморфности может потребовать много компьютерного времени. Это одна из так называемых *NP-трудных* проблем в теории сложности решений [9].

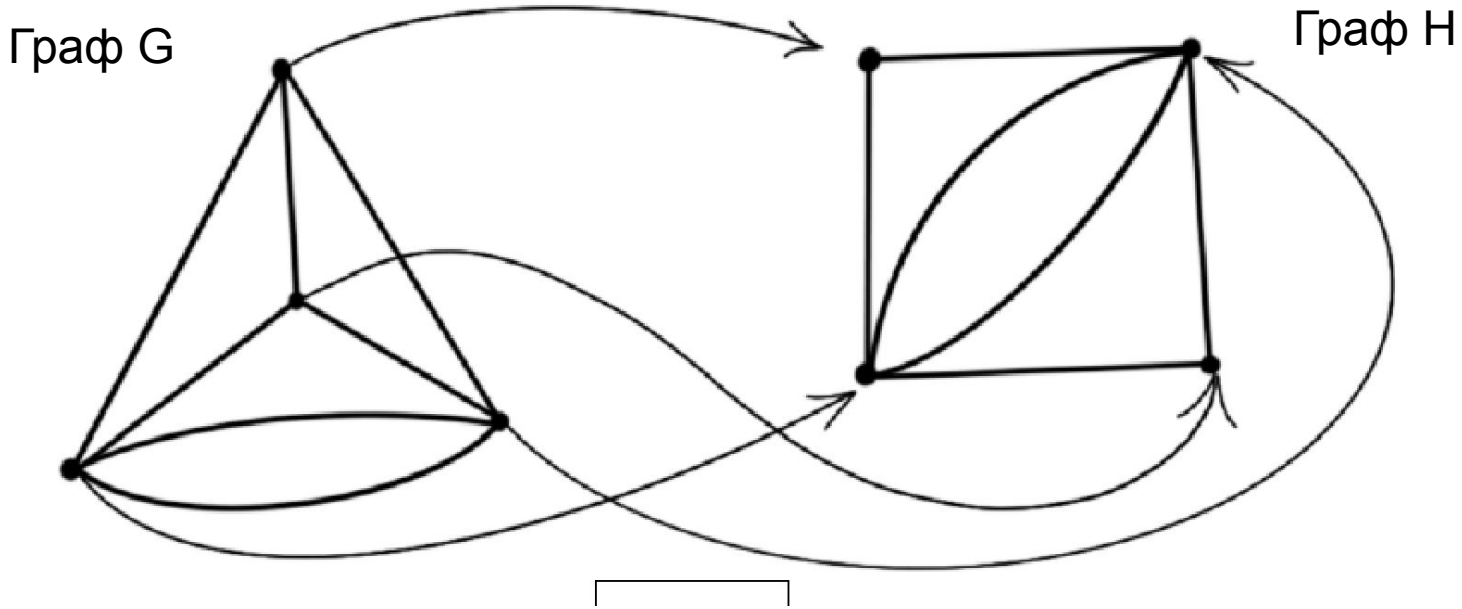
Предположим, что некоторый граф *G* известен как P, так и V, пусть также P удалось каким-то образом найти на нем ГЦ и он хочет доказать этот факт V, не выдавая ГЦ.

Задача коммивояжера



Задача коммивояжера –обойти все узлы (города), побывав в каждом Только 1 раз. Кратчайший путь называется гамильтоновым циклом.

Изоморфизм графов



Доказывающий показывает, что

1. Либо новый граф изоморфен исходному.
2. Либо решение задачи на новом графе

Протокол доказательства с нулевым разглашением принимает для данной задачи следующий вид:

1. P случайным образом переставляет нумерацию вершин графа *G*, получая другой граф *H*, который будет, очевидно, изоморфен *G*.

Так как P знает этот изоморфизм и ему известен ГЦ на графе *G*, он легко находит такой же ГЦ на графе *H*. С другой стороны, если бы P не знал ГЦ на графе *G*, то он не смог бы в обозримое время найти его и на графе *H*. Более того, если бы P знал ГЦ на каком-то другом графе , то он не смог бы доказать в обозримое время, что и *G* топологически изоморфны (даже если бы это было верно), поскольку доказательство этого факта для произвольных графов является, как уже отмечалось ранее, трудной задачей. (Две необозримо трудные задачи)

2. P посылает V копию графа *H*.
3. V просит P выполнить **одно из двух**:
а) доказать, что *H* и *G* изоморфны,
б) показать ГЦ на *H*.
4. P исполняет его просьбу, делает **одно из двух**:
а) доказывает, что *H* и *G* изоморфны, не показывая ГЦ на *G* или *H*,
б) показывает ГЦ только на *H*, не доказывая изоморфизма *G* и *H*.
5. P и V повторяют *n* раз шаги протокола 1–4.

Если P знает ГЦ на графе *G*, то он сможет правильно выполнить задания V на каждой из *n* итераций.

Если P этого не знает, то он не сможет выполнить требования V на всех шагах. Лучшее, что сможет сделать нечестный P, это построить такой граф , который будет или изоморфным *G*, или имеющим известный ему ГЦ. Очевидно, что у P оказывается только 50% шансов угадать, какое доказательство потребует от него V на шаге 3. Таким образом, задавая достаточно большое количество итераций *n*, можно надежно обеспечить разоблачение нечестного P.

С другой стороны, этот протокол не дает V никакой информации, помогающей ему из ответов P установить ГЦ графа *G*, поскольку P для каждого нового раунда протокола генерирует новый граф *H*.

Видно, что протокол, рассмотренный выше, требует обмена информацией между P и V, поэтому такого типа протоколы называются *интерактивными.*

Но протоколы с нулевым разглашением не обязательно должны быть интерактивными. Это означает, что P публикует все необходимые данные заранее и каждый пользователь имеет возможность проверить, что P обладает некоторыми секретными знаниями, не получив из этих знаний никакой информации и не вступая даже в диалог с P!
Описание *неинтерактивного* протокола для доказательства ГЦ заданного графа можно найти в [15].

Идентификация (аутентификация) пользователей при помощи протокола с нулевым разглашением

Широкое распространение интеллектуальных карт для разнообразных коммерческих, гражданских и военных применений потребовало обеспечения безопасности идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать нарушителю в допуске, ответе или обслуживании.

Способы:

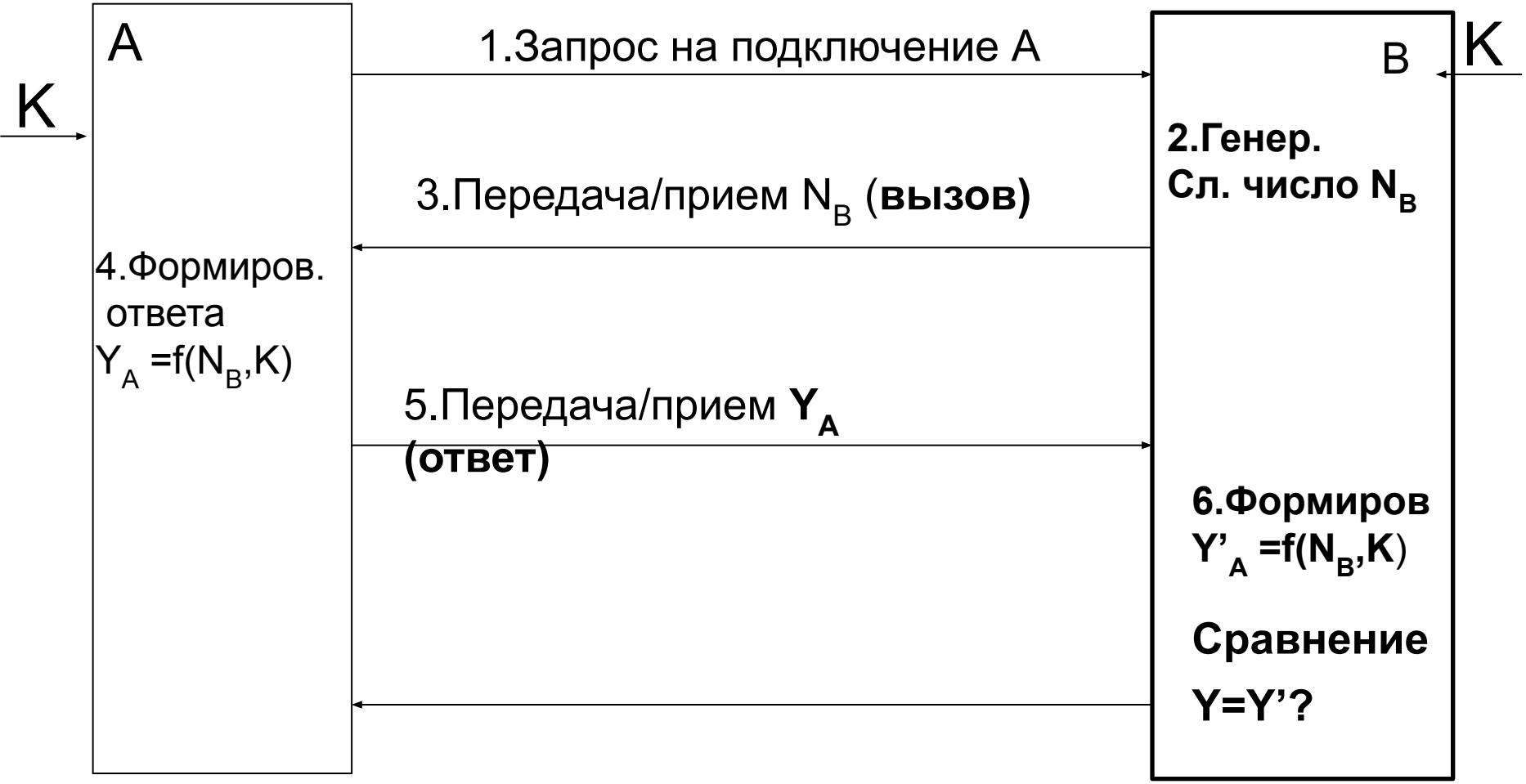
1. Схемы идентификации на **основе паролей** слабо соответствуют требованиям указанных приложений. Один из существенных недостатков такой идентификации заключается в том, что после того, как доказывающий передаст проверяющему пользователю свой пароль, проверяющий может, используя данный пароль, выдать себя впоследствии за проверяемого пользователя.

- 2**. Протоколы идентификации на основе симметричных алгоритмов шифрования**
- Недостатки. Для работы таких протоколов идентификации необходимо, чтобы проверяющий и доказывающий с самого начала имели один и тот же секретный ключ.
  Следовательно, встает вопрос о распределении и доставке секретных ключей.

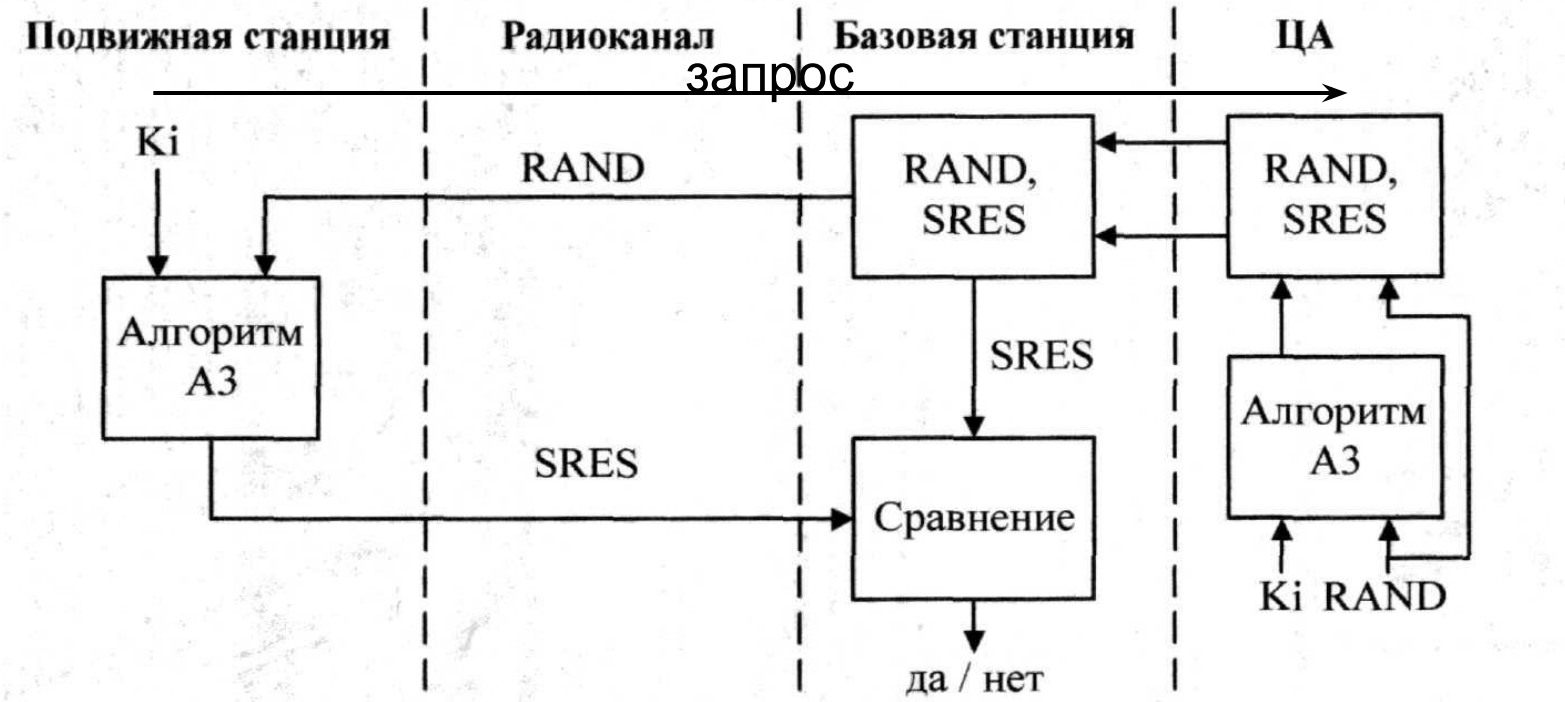
  При исполнении таких протоколов пользователь доказывает знание секретного ключа, производя с его помощью расшифрование запросов. Проверяющий пользователь имеет принципиальную возможность так сформировать запросы, чтобы передаваемые ответы могли быть обработаны в целях извлечения из них дополнительной информации о секретном ключе с последующим возможным раскрытием этого ключа [15]

2. **Протоколы идентификации на основе симметричных алгоритмов шифрования**

Аутентификация способом вызов-ответ



Аутентификация пользователя в стандарте мобильной связи



3. Способы идентификации на основе использования цифровой подписи

В случае идентификации на основе ЦП в ней присутствует информация о секретном ключе подписи, которую, хотя и вычислительно сложно, но принципиально можно найти.

Кроме того, алгоритмы выполнения и проверки ЦП содержат в себе сложные операции модульного возведения в степень больших чисел, требующие при их программной реализации больших ресурсов процессорного времени, тогда как в алгоритме идентификации с нулевым разглашением (НР) применяются гораздо более простые модульные математические операции (возведение в квадрат и умножение), что позволяет значительно снизить требования к вычислительным ресурсам верификации.

4. Идентификация пользователей на основе протокола с нулевым разглашением

Преимущество идентификации на основе использования доказательств с нулевыми знаниями над остальными способами идентификации (в частности, и на основе ЦП) заключается в том, что в ходе ее выполнения никакой информации о секретном ключе не «утекает» к проверяющему и ко всем посторонним наблюдателям.

Рассмотрим далее пример построения подобного протокола [15].

Пусть секретным ключом пользователя будет цепочка чисел $\overline{C} = (C_1, C_2, \ldots, C_k)$, где $1 \le C_j \le n$, а $n=pq$ – модуль, p и q большие простые числа, причем координаты $\overline{C}$ удовлетворяют уравнениям

$$d_j \cdot C_j^2 = (\pm 1) \bmod n,\ i = 1, 2, 3, \ldots, k \qquad (5.1)$$

где $d_1, d_2, \ldots, d_k$ – открытый идентификатор пользователя.

Некоторый центр предварительно формирует число $n=pq$, где p и q – большие простые числа, причем p=3mod4 и q=3mod4; далее необходимость в обращении к центру для выполнения протокола идентификации отпадает.

Проверяющий (V) знает число n и то, что секретный идентификатор определенного пользователя $\overline{C}$ удовлетворяет уравнению (5.1), но сам этот идентификатор для него неизвестен.

При выполнении протокола идентификации проверяющий V по предъявленным ему данным должен убедиться, что их автор имеет в своем распоряжении $\overline{C}$, но не может получить больше об этом векторе никакой *дополнительной информации*.

Однако вычислительная мощность V должна быть ограничена, поскольку в противном случае V сможет найти $\overline{C}$ еще до выполнения протокола и, следовательно, выдавать себя в дальнейшем за пользователя P.

Протокол HP состоит из четырех шагов:

1. Проверяемый пользователь (P) генерирует случайное число *r* и находит число

$$x = r^2 \bmod n \tag{5.2}$$

Далее он посылает это число V.

2. V генерирует случайное подмножество $S \subseteq (1, 2, \ldots, k)$ и отправляет его P.

3. P вычисляет

$$T_c = \prod_{j \in S} C_j \bmod n \tag{5.3}$$

и посылает V число

$$y = r \cdot T_c \bmod n \tag{5.4}$$

4. V вычисляет

$$X = y^2 \cdot T_d \bmod n \tag{5.5}$$

где

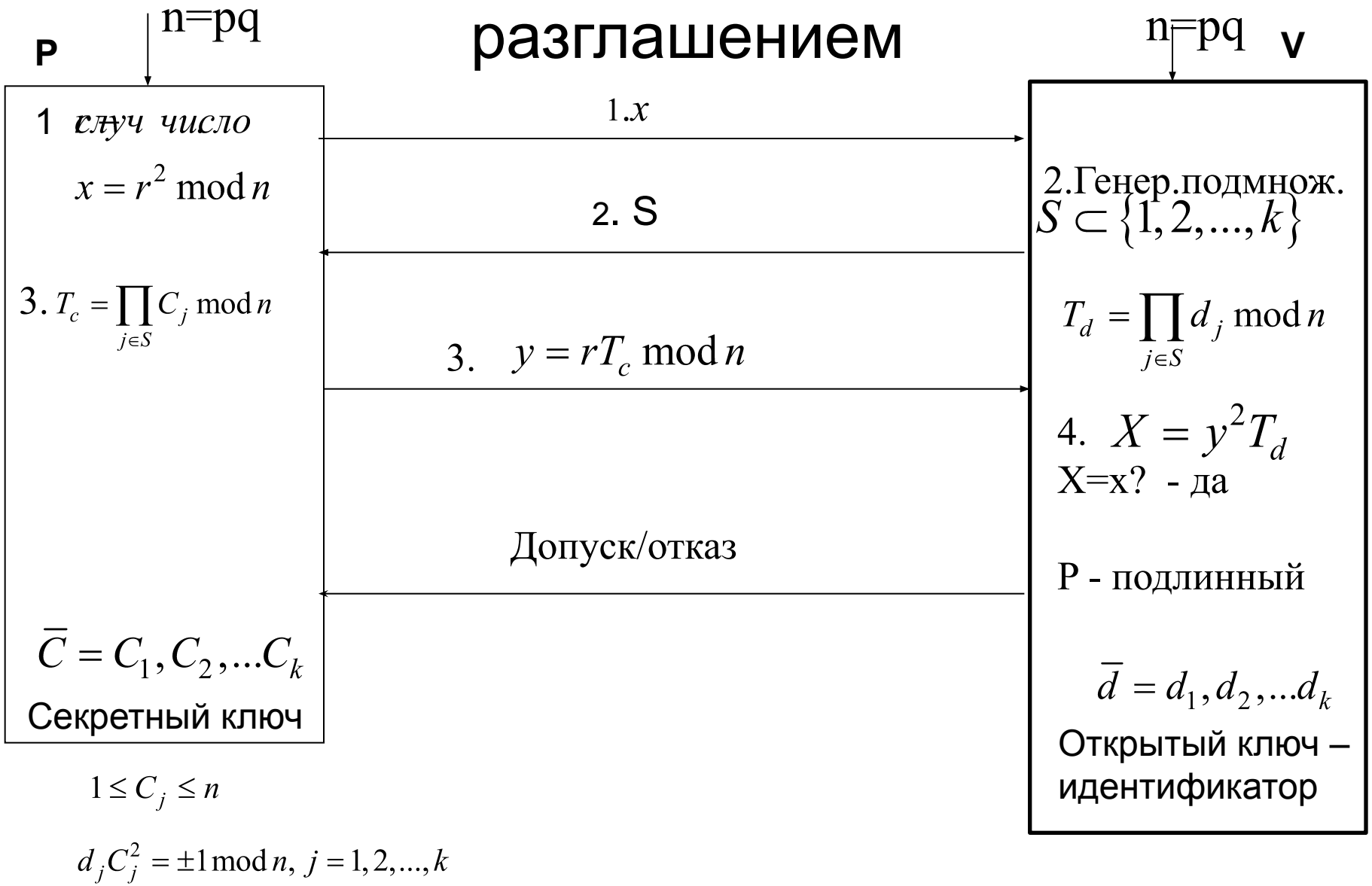
$$T_d = \prod_{j \in S} d_j \bmod n \tag{5.6}$$

и проверяет равенство

$$x = \pm X \bmod n \tag{5.7}$$

Если равенство (5.7) выполняется, то производится следующая итерация (т. е. повторение протокола с новыми случайными данными). Если равенство (5.7) не выполняется, то проверяемый пользователь P не идентифицируется V.

Аутентификация на основе доказательства с нулевым разглашением



Если справедливо равенство (5.1) (т. е. P является действительно обладателем секретного вектора $\overline{C}$), то уравнение (5.7) всегда выполняется. Действительно,

$$y^2 \cdot T_d = r^2 \cdot T_c^2 \cdot T_d = \pm r^2 = \pm x \bmod n \qquad (5.8)$$

Умножение случайного числа *r* на T_c на шаге 3 действительно необходимо, поскольку иначе, выбирая на каждой итерации подмножество $S = \{j\}$, т. е. на первой итерации – $S = \{1\}$, на второй – $S = \{2\}$, и так далее до $S = \{k\}$, V сможет вычислить весь секретный идентификатор $C_1, C_2, \ldots, C_k$

Стойкость данной системы идентификации базируется на невозможности извлечения квадратных корней по mod*n* при неизвестной факторизации *n*. Никакой другой информации о векторе $\overline{C}$ проверяющий V в процессе выполнения протокола не получает. (Предполагается, что всегда выполняется условие $\gcd(C_j, n) = 1$, поскольку в противном случае модуль *n* может быть легко факторизован.)

С другой стороны, существует только один способ для P обмануть V, т. е. обеспечить себе идентификацию при отсутствии у него чисел C_j. Этот способ заключается в угадывании подмножества *S* и формировании $x = \pm r^2 \cdot T_d \bmod n$ и *y*=*r*. Тогда равенство $x = y^2 \cdot T_d \bmod n$ будет всегда тривиально выполняться.

Вероятность угадывания множества *S* со стороны P равна 2^{-k}, и тогда вероятность обмана при выполнении *t* итераций будет равна $2^{-k \cdot t}$. Соответственно чем больше число таких итераций, тем меньше вероятность обмана со стороны P при выполнении такого протокола.

Заметим также, что дополнительные условия на простые множители *n* нужны для того, чтобы V мог быть уверен, что числа C_j, соответствующие числам d_j, существуют.

Пример выполнения процедуры идентификации.

Пусть центр предоставил пользователю модуль *n*=19·31=589. Предположим, что идентифицируемый пользователь сгенерировал секретный идентификатор $\overline{C} = (90, 544, 460, 263, 567)$ Открытый идентификатор *d(p)* вычисляется как решение уравнения (5.1). Это уравнение решается путем нахождения обратных элементов к c_j^2 по модулю *n*, что дает, как легко проверить, вектор $\overline{d} = (472, 121, 253, 283, 359).$

Выполним одну итерацию идентификации:

1. P генерирует случайное число *r*=859, вычисляет $x = 859^2 \bmod 589 = 453$ и посылает его V.

2. V генерирует множество *S*=(1,5,4) и отправляет его P.

3. P вычисляет $T_c = \prod_{j \in S} C_j = 90 \cdot 567 \cdot 263 = 13420890$ и посылает V число $y = 859 \cdot T_c \bmod 589 = 390$

4. V находит $T_d = \prod_{j \in S} d_j = 472 \cdot 359 \cdot 283 = 47953784$, затем вычисляет $X = 390^2 \cdot 47953784 \bmod 589 = 453$ и проверяет равенство (5.7): *x*=*X*. Так как последнее равенство выполнилось, *V* делает вывод об успешной идентификации P.

Достоинство протокола идентификации с нулевым разглашением (НР) по сравнению с алгоритмом ЭЦП, в том что в нем применяются гораздо более простые модульные математические операции (возведение в квадрат и умножение), что позволяет значительно снизить требования к вычислительным ресурсам верификации. Однако, как и в ЭЦП необходимо гарантировать принадлежность открытого ключа (последовательности чисел d1, d2,…..) ее владельцу, что как правило решается с помощь сертификатов.