Задание №5

АЛГОРИТМЫ ИДЕНТИФИКАЦИИ

В своих работах [544, 545], Фейге, Фиат и Шамир показали, как параллельная схема может повысить число аккредитаций на этап и уменьшить взаимодействия Пегги и Виктора .

Сначала, как и в предыдущем примере, генерируется n, произведение двух больших простых чисел. Для генерации открытого и закрытого ключей Пегги сначала выбирается k различных чисел: $v_1, v_2, \ldots v_k$, где каждое v_i является квадратичным остатком mod n. Иными словами, v_i выбираются так, чтобы $x^2 \equiv v_i \pmod n$ имело решение, и существовало $v_i^{-1} \bmod n$. Строка, $v_1, v_2, \ldots v_k$, служит открытым ключом. Затем вычисляются наименьшие s_i, для которых $s_i \equiv \text{sqrt}\,(v_i^{-1}) \pmod n$. Строка $s_1, s_2, \ldots s_k$, служит закрытым ключом.

Выполняется следующий протокол:

(1) Пегги выбирает случайное r, меньшее n. Затем она вычисляет $x = -r^2 \bmod n$ и посылает x Виктору.

(2) Виктор посылает Пегги строку из k случайных битов: $b_1, b_2, \ldots b_k$.

(3) Пегги вычисляет $y = r * (s_1^{b_1} * s_2^{b_2} * \ldots * s_k^{b_k}) \bmod n$. (Она перемножает вместе значения s_i, соответствующие b_i=1. Если первым битом Виктора будет 1, то s_1 войдет в произведение, а если первым битом будет 0, то нет, и т.д.) Она посылает y Виктору.

(4) Виктор проверяет, что $x = y^2 * (v_1^{b_1} * v_2^{b_2} * \ldots * v_k^{b_k}) \bmod n$. (Он перемножает вместе значения v_i, основываясь га случайной двоичной строке. Если его первым битом является 1, то v_1 войдет в произведение, а если первым битом будет 0, то нет, и т.д.)

Пегги и Виктор повторяют этот протокол t раз, пока Виктор не убедится, что Пегги знает $s_1, s_2, \ldots s_k$.

Вероятность, что Пегги удастся обмануть Виктор t раз, равна $1/2^{kt}$. Авторы рекомендуют использовать вероятность мошенничества $1/2^{20}$ и предлагают значения $k = 5$ и $t = 4$. Если у вас склонность к мании преследования, увеличьте эти значения.

Пример

Взглянем на работу этого протокола небольших числах. Если $n = 35$ (два простых числа - 5 и 7), то возможными квадратичными остатками являются:

1: $x^2 \equiv 1 \pmod{35}$ имеет решения: $x = 1, 6, 29, 34$.

4: $x^2 \equiv 4 \pmod{35}$ имеет решения: $x = 2, 12, 23, 33$.

9: $x^2 \equiv 9 \pmod{35}$ имеет решения: $x = 3, 17, 18, 32$.

11: $x^2 \equiv 11 \pmod{35}$ имеет решения: $x = 9, 16, 19, 26$.

14: $x^2 \equiv 14 \pmod{35}$ имеет решения: $x = 7, 28$.

15: $x^2 \equiv 15 \pmod{35}$ имеет решения: $x = 15, 20$.

16: $x^2 \equiv 16 \pmod{35}$ имеет решения: $x = 4, 11, 24, 31$.

21: $x^2 \equiv 21 \pmod{35}$ имеет решения: $x = 14, 21$.

25: $x^2 \equiv 25 \pmod{35}$ имеет решения: $x = 5, 30$.

29: $x^2 \equiv 29 \pmod{35}$ имеет решения: $x = 8, 13, 22, 27$.

30: $x^2 \equiv 30 \pmod{35}$ имеет решения: $x = 10, 25$.

Обратными значениями (mod 35) и их квадратными корнями являются:

v	v^{-1}	s=sqrt(v^{-1})
1	1	1
4	9	3
9	4	2
11	16	4

16	11	9
29	29	8

Обратите внимание, что у чисел 14, 15, 21, 25 и 30 нет обратных значений mod 35, так как они не взаимно просты с 35. Это имеет смысл, так как должно быть (5 - 1) * (7 - 1)/4 квадратичных остатков mod 35, взаимно простых с 35: НОД(x, 35) = 1 (см. раздел 11.3).

Итак, Петти получает открытый ключ, состоящий из k = 4 значений: {4,11,16,29}. Соответствующим закрытым ключом является {3,4,9,8}. Вот один этап протокола.

(1) Петти выбирает случайное r=16, вычисляет $16^2 \bmod 35 = 11$ и посылает его Виктору.

(2) Виктор посылает Петти строку случайных битов: {1, 1, 0, 1}

(3) Петти вычисляет $16*(3^1*4^1*9^0*8^1) \bmod 35 = 31$ и посылает его Виктору.

(4) Виктор проверяет, что $31^2*(4^1*11^1*16^0*29^1) \bmod 35 = 11$.

Петти и Виктор повторяют этот протокол t раз, каждый раз с новым случайным r, пока Виктор будет убежден.

Небольшие числа, подобные использованным в примере, не обеспечивают реальной безопасности . Но когда длина n равна 512 и более битам, Виктор не сможет узнать о закрытом ключе Петти ничего кроме того факта, что Петти действительно знает его.

Схема подписи Fiat-Shamir

Превращение этой схемы идентификации в схему подписи - это, по сути, вопрос превращения Виктора в хэш-функциию. Главным преимуществом схемы цифровой подписи Fiat-Shamir по сравнению с RSA является ее скорость: для Fiat-Shamir нужно всего лишь от 1 до 4 процентов модульных умножений, используемых в RSA. В этом протоколе снова вернемся к Алисе и Бобу.

Смысл переменных - такой же, как и в схеме идентификации. Выбирается n - произведение двух больших простых чисел. Генерируется открытый ключ, $v_1, v_2, \ldots v_k$, и закрытый ключ, $s_1, s_2, \ldots s_k$, где $s_i \equiv \text{sqrt}\ (v_i^{-1}) \pmod n$.

(1) Алиса выбирает t случайных целых чисел в диапазоне от 1 до n - $r_1, r_2, \ldots, r_t$ - и вычисляет $x_1, x_2, \ldots x_t$, такие что $x_i = r_i^2 \bmod n$.

(2) Алиса хэширует объединение сообщения и строки x_i, создавая битовый поток: $H(m, x_1, x_2, \ldots x_t)$. Она использует первые $k*t$ битов этой строки в качестве значений b_{ij}, где i пробегает от1 до t, а j от 1 до k.

(3) Алиса вычисляет $y_1, y_2, \ldots y_t$, где $y_i = r_i * (s_1^{b_{i1}} * s_2^{b_{i2}} * \ldots * s_k^{b_{ik}}) \bmod n$

(Для каждого i она перемножает вместе значения s_i, в зависимости от случайных значений b_{ij}. Если b_{ij}=1, то s_i участвует в вычислениях, если b_{ij}=0, то нет.)

(4) Алиса посылает Бобу m, все биты b_{ij}, и все значения y_i. У Боба уже есть открытый ключ Алисы: $v_1, v_2, \ldots v_k$.

(5) Боб вычисляет $z_1, z_2, \ldots z_t$, где $z_i = y^2 * (v_1^{b_{i1}} * v_2^{b_{i2}} * \ldots * v_k^{b_{ik}}) \bmod n$

(И снова Боб выполняет умножение в зависимости от значений b_{ij}.) Также обратите внимание, что z_i должно быть равно x_i.

(6) Боб проверяет, что первые $k*t$ битов $H(m, z_1, z_2, \ldots z_t)$ - это значения b_{ij}, которые прислала ему Алиса.

Как и в схеме идентификации безопасность схемы подписи пропорциональна $1/2^{kt}$. Она также зависит от сложности разложения n на множители. Фиат и Шамир показали, что подделка подписи облегчается, если сложность разложения n на множители заметно меньше 2^{kt}. Кроме того, из-за вскрытия методом дня рождения

З а д а н и е .

1. Ознакомиться с алгоритмом идентификации.
2. Разработать программу идентификации.
3. Произвести контрольную идентификацию.
4. Оценить затраченные ресурсы компьтера.
5. Оценить затраченные ресурсы и время.