Задача

Вариант 12.

Реализовать алгоритм электронной цифровой подписи DSA.

Требования к работе:

- При постановке подписи использовать хеш-функцию MD5.
- Реализовать атаку на систему DSA в случае, если стал известен случайный параметр k.

Тестируемый алгоритм

DSA представляет собой криптографический алгоритм с использованием закрытого ключа для создания электронной подписи, но не для шифрования. Подпись создается секретно (закрытым ключом), но может быть публично проверена (открытым ключом). Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность. Алгоритм основан на вычислительной сложности взятия логарифмов в конечных полях.

Описание алгоритма

DSA включает в себя два алгоритма (S, V): для создания подписи сообщения (S) и для ее проверки (V).

Оба алгоритма вначале вычисляют хеш сообщения, используя криптографическую хеш-функцию MD5. Алгоритм S использует хеш и секретный ключ для создания подписи, алгоритм V использует хеш сообщения, подпись и открытый ключ для проверки подписи.

Стоит подчеркнуть, что фактически подписывается не сообщение (произвольной длины), а его хеш (128 бит), поэтому неизбежны коллизии и одна подпись, вообще говоря, действительна для нескольких сообщений с одинаковым хешем.

Параметры схемы цифровой подписи:

1. Хеш-функция – MD5
2. Простое число q размерностью 128 бит
3. Простое число p, такое, что (p-1) делится на q
4. Число $g = 2^{(p-1)/q}$

5. Секретный ключ x ϵ (0;q)
6. Открытый ключ $y = g^x mod\ p$

Подпись сообщения m:

1. Выбор случайного числа k ϵ (0;q)
2. Вычисление $r = (g^k mod\ p) mod\ q$
3. Выбор другого k, если r = 0
4. Вычисление $s = k^{-1}(MD5(m) + x * r)\ mod\ q$
5. Выбор другого k, если s = 0
6. Подписью является пара (r, s) общей длины 2N

Проверка подписи:

1. Вычисление $w = s^{-1}\ mod\ q$
2. Вычисление $u_1 = MD5(m) * w\ mod\ q$
3. Вычисление $u_2 = r * w\ mod\ q$
4. Вычисление $v = (g^{u_1} * y^{u_2}\ mod\ p)\ mod\ q$
5. Подпись верна, если v=r

Пример использования алгоритма

Исходное сообщение: «sign»

```
q = 27006903867799759203369573102506443447 7
p = 13166691828078804146195296381046017440853320051791679313527650100635102845257698288744364711355938801543913644020125646474073661196849815934677298670992467
g = 12238650980896238355305173443595273376825031421316715135079056635838355209037557517099173190765679608853782684270357750402200333000975846277739060160153955
open key = 13319573691392384134186139155831347286544130126315377061364509961143581324673091953183653297167666141333445171709627255549984331848423961750769670889416 71
private key = 71670676000728878261723215687697050 36
sign 25303170047018428855024867730863205946 8 157344976583155211942179495682954414480
```

Проверим подпись:

```
sign is correct
```

Изменим подпись и проверим ее:

```
sign 25303170047018428855024867730863205946 9 157344976583155211942179495682954414480
sign is not correct
```

Атака

Когда случайный параметр к становится известен, узнать секретный ключ становится достаточно

просто:

$xr \ mod \ q = ks - MD5(m) \ mod \ q$

```
finded x = 716706760007288782617232156876970503 6
```

Вывод

В данной лабораторной работе был реализован алгоритм цифровой подписи DSA, по которому была проведена атака при известном секретном ключе к. Тем не менее, если все секретные параметры остаются секретными, взломать данный алгоритм не так просто, а его стойкость увеличивается с увеличением длины хеша и простых чисел p и q.