РУТОКЕН

Особенности реализации новых российских криптографических стандартов на процессорах архитектуры ARM7

Тараскин Олег Геннадьевич

компания АКТИВ

Криптографические стандарты:

Старые криптографические госты:

ГОСТ Р 34.10-2001 - ЭЦП

ГОСТ Р 34.11-94 - хэш

ГОСТ 28147- 89 - шифрование

Новые :

ГОСТ Р 34.10-2012 - ЭЦП

ГОСТ Р 34.11-2012 - хэш

? - шифрование

Другие полезные документы

Самый интересный документ :

RFC 4357 : Additional Cryptographic Algorithms for Use with GOST 28147- 89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms

Содержит:

1. узлы замен (S-box) для ГОСТ 28147-89,
2. параметры эллиптических кривых (domain parameters),
3. Алгоритм VKO GOST R 34.10-2001 – “наш” вариант ECC Diffie-Hellman
4. многое другое

Основная цель RFC 4357 – совместимые реализации ГОСТов

RFC 4490: Using the GOST 28147-89, GOST R 34.11-94,

GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS)

ГОСТ Р 34.10-2012

Эллиптическая кривая над GF(p), p – простое > 3 - мн-во пар чисел (x, y) - точек из GF(p): $y^2 \equiv x^3 + a*x + b \pmod p$ *и Точка на бесконечности (Point at infinity).*

Где $4*a^3+27*b^2$ *не сравн. с 0* (mod p)

Групповой закон сложения точек:

(x3, y3) = (x1, y1) + (x2, y2) :

(x1 != x2)

$x3 \equiv L^2 - x1 - x2 \pmod p$

$y3 \equiv L*(x1-x3) - y1 \pmod p$

, где $L = (y2-y1)/(x2-x1) \pmod p$

ГОСТ Р 34.10-2012

Если x1 = x2 и y1 =y2 != 0 , то *(x3, y3) :*

x3 ≡ L^2 – 2*x1 (mod p)

y*3* ≡ L*(x1-x3) – y1 (mod p)

, где L = (3*x1^2 + a)/(2*y1^2) (mod p)

Если же x1 = x2 и y1 ≡ -y2 (mod p) ,

то *(x3, y3) – точка на бесконечности O.*

Скалярное умножение точки P на число k :

*точка Q = P + … + P (к раз) = k*P*

ГОСТ Р 34.10-2012

Формирование подписи:

1. Выч. хэш h от сообщения
2. Выч. $e \equiv h \pmod q$, где q – порядок подгруппы
   если e = 0, то опред. e = 1
3. Ген. случ. число k : $0 < k < q$
4. Вычисл. точку $C = k*P$, P – базовая точка подгруппы
   $r \equiv Xc \pmod q$, где Xc - x коорд. точки C
   если r = 0, то шаг 3
5. Выч. $s \equiv (r*d + k*e) \pmod q$, где d – закрытый ключ
   если s = 0, то шаг 3

Результат : (r, s)

SPA(simple power analysis)

Общая схема атаки:

1. Исследуется энергопотребление при операции k*P. При успехе атакующий получает k.
2. Имея подпись (r, s) , получаем линейное сравнение с одним неизвестным d:

   $s \equiv (r*d + k*e) \pmod{q}$

SPA(simple power analysis)

Бинарный алгоритм “Double and add”

Вход : точка *P и n-битное число k*

Выход : точка C = k*P

1. $Q \leftarrow P$
2. **for** i = (n − 2) **down to** 0 **do**

```
{
    Q ← [2]Q
    if ( бит i == 1)
        Q ← Q + P     -   утечка
}
```

3. **return** *Q*

SPA(simple power analysis)

Бинарный алгоритм “Double and add always”

Вход : точка *P и n-битное число k*

Выход : точка C = k*P

1. $Q \leftarrow P$
2. **for** i = (n − 2) **down to** 0 **do**

```
{
    Q0 ← [2]Q
    Q1 ← Q0 + P
    if ( бит i == 1)
        Q ← Q1
    else
        Q ← Q0
}
```

3. **return** *Q*

Итого: SPA(simple power analysis)

Возможен из-за того, что для сложения точек требуется больше энергии, чем для удвоения.

При “наивной” реализации k*P, т.е. без вставки в алгоритм dumb – операций сложения точек данная атака может быть применена как к ARM 7 так и к защищенным криптосопроцессорам.

Для ее осуществления требуется всего одно измерение потребляемой мощности k*P.

DPA(differential power analysis)

Имеет смысл применять, только если устройство защищено от SPA.

Требует по порядку от нескольких сотен измерений.

Данный метод анализа неприменим против ГОСТ Р 34.10-2012, т.к. при k*P каждый раз используется новое k.

Применим к VKO, если у атакующего есть не только возможность измерять энергию, но и навязывать устройству входные значения алгоритма.

DPA(differential power analysis)

VKO :

K = ((UKM*PrivA)(mod q)) *PubY

KEK = Hash(K)

Атакующий будет каждый раз посылать один и тот же UKM -> скаляр (UKM*PrivA)(mod q) будет const

Контрмеры:

Рандомизация точки и скаляра

DPA(differential power analysis)

Рандомизация представления точки :

Projective coordinates (X, Y, Z) :

Y^2 * Z ≡ X^3 + a*X*Z^2 + b*Z^3 (mod p), Z != 0
(X, Y, Z) -> (X/Z, Y/Z)
Сложение : 12M + 2S, удвоение : 7M + 5S
Рандомизация :
Ген. случ. R !=0 и (X, Y, Z) -> (R*X, R*Y, R*Z)

Jacobian coordinates (X, Y, Z) :

Y^2 ≡ X^3 + a*X*Z^4 + b*Z^6 (mod p) , Z != 0
(X, Y, Z) -> (X/Z^2, Y/Z^3)
Сложение : 12M + 4S, удвоение : 4M + 6S
Рандомизация :
(X, Y, Z) -> ((R^2)*X, (R^3)*Y, R*Z)

DPA(differential power analysis)

Рандомизация скаляра :

- Blinding: k*P = (k + r*ord(P))*P
- Additive splitting: k = k1 + k2,

где k1 = r, k2 = k – r

-> k*P = k1*P + k2*P (длина r порядка длины k)

- k = k1*r + k2, где k1 = k/r , k1 = k mod r

->k*P = k1*(r*P) +k2*P

- Mult. splitting: k*P = (k*r^(-1))*(r*P)

Fault attack’s

для VKO:

- 1. Не выпускать из устройства точку или
- 2. В VKO проверять результат скалярного умножения
- А лучше и 1 и 2

Оптимизация вычислений по mod P

- Кривая с модулем общего вида:
  1. Арифметика Montgomery
  2. Арифметика Quisquater’a (патент у NXP)
- Кривая со специальным модулем :

  (Например: p = 2^512 – 569

  или p = 2^511 + 111)

  Простое умножение и вычисление mod P

Результаты для Rutoken ECP

- Время подписи ГОСТ 34.10-2001 с параметрами CryptoPro ParamSet A (модуль P = 2^256 - 617) равно 0, 41 сек.
- Время подписи ГОСТ 34.10-2012 с (модуль P = 2^512 -) равно 1, 6 сек.

Контактная информация

Тараскин Олег Геннадьевич
Email:
tog@rutoken.ru

www.rutoken.ru