# *Криптоанализ RSA*

Докладчик: Николай Гравин

(311 Группа)

# RSA:

- Берем p,q- два больших простых числа(512 бит)
- $n=p\text{Ч}q$, $\phi(n)=(p-1)\text{Ч}(q-1)$
- $e<\phi(n)$ ,такое что $gcd(e, \phi(n))=1$
- d-? : $e\text{Ч}d=1 \pmod{\phi(n)}$
- (e,n)-открытый ключ, d-закрытый ключ

Задача:

- Как зная e и $\phi(n)$ найти за полиномиальное время такое d(такое что $e\text{Ч}d=1 \pmod{\phi(n)}$).

# *Encryption and Digital Signature*

- Шифрование:
- $MOZ_n$(секретное сообщение)
- $C=M^e \pmod n$ то, что мы посылаем получателю.
- $D=C^d \pmod n$ $D=M$, $D$ является расшифровкой $C$

- Цифровая подпись:
- M-сообщение или Hash от него
- Мы посылаем $(M,S)$, где $S=M^d \pmod n$–подпись.
- Каждый может проверить, что $S^e=M$, но не может сам придумать по $M$ такое $S$.

# *Полезный теоретический факт*

- Пусть (N,e)-публичный ключ, d- закрытый ключ. Тогда зная (N,e,d) можно разложить N на простые множители N=pЧq за полиномиальное время.

# Полезный теоретический факт

- Пусть (N,e)-публичный ключ, d- закрытый ключ. Тогда зная (N,e,d) можно разложить N на простые множители N=pЧq за полиномиальное время.

- Задача:
  - Докажите этот факт.

# Теоретический факт

- ***<u>Открытый вопрос</u>***: Пусть даны N,e:gcd(e,ϕ(n))=1 и F:Zn->Zn, F(x)=x^(1/e)(mod n) – вычисляется за единичное время. Существует ли тогда полиномиальный алгоритм, раскладывающий N на простые множители.(F(x)-’оракул’)
- Результат: для малых ***e*** ответ нет. Boneh и Venkatesan доказали,что в определенной модели, ответ ‘Да’ на вопрос для малых ***e*** даст нам эффективный алгоритм разложения N.

# Методы разложения N на простые сомножители

- Trial Division
- Pollard’s p-1 Method
- Pollard’s rho Method
- Elliptic Curve Method
- Quadratic Sieve Method
- Number Field Sieve Method

# Trial Division

- Пытаемся разделить n на все простые числа от 1 до Цn.

# Trial Division

- Пытаемся разделить n на все простые числа от 1 до Цn.
- Плохой метод (работает log(n)*2n^1/2)

# Trial Division

- Пытаемся разделить n на все простые числа от 1 до Цn.
- Плохой метод (работает log(n)*2n^1/2)
- Хороший метод так, как больше чем у 91% чисел есть простой делитель меньший 1000.

# Pollard's p-1 Method

- n=pq , у p-1 все простые делители <B
- k- произведение достаточно больших степеней всех простых чисел <B, тогда p-1|k.
- Пусть a=2. p|(a^k-1), значит мы можем найти p, как gcd(n, (a^k-1)).

# Pollard's rho(ρ) Method

- Если у нас есть n исходов и 1.2*(n^1.2) испытаний то вероятность того, что 2 элемента совпали >50%.(birthday paradox)
- Теперь придумаем какую-нибудь функцию f: $Z_n \to Z_n$, которая ведет себя в $Z_n$ 'рандомно'(f(x)=x^2+1(mod n)-подойдет)
- Начнем выписывать последовательность $x_1, x_2, x_3, \ldots$ , где $x_{i+1}=f(x_i)$, параллельно будем считать $\gcd(x_i-x_j,n)$ для всех i и j – если gcd не 1 то мы разложили n.

# Pollard’s rho(ρ) Method

- <u>Замечание</u> Если считать для всех пар i и j gcd($x_j$-$x_i$,n), то мы сделаем слишком много операций.

# Pollard's rho(ρ) Method

- Замечание Если считать для всех пар i и j gcd($x_j$-$x_i$,n), то мы сделаем слишком много операций.
- Вопрос: Как этого избежать?

# Pollard's rho(ρ) Method

- <u>Замечание</u> Если считать для всех пар i и j gcd($x_j$-$x_i$,n), то мы сделаем слишком много операций.
- Вопрос: Как этого избежать?
- Ответ: Проверять только для j=2i.

# *Литература*

- Twenty Years of Attacks on the RSA Cryptosystem (Dan Boneh)
- The Quadratic Sieve Factoring Algorithm (Eric Landquist)
- Cryptanalysis of RSA: A Survey (Calros Frederico Cid)