



Cryptology Lab assignment 2: Making and breaking RSA

Vasilij Savin

Information Technology Department
Uppsala University

Spring 2009



Lab deliverables

- Work groups consist of 2-3 students
- Laboration dates (room 1515D):
 - ✿ April 28 – 8:00 -17:00
- Deadline: Monday, May 4th
- Examination sessions: Friday, May 8th, 8:00-12:00
- Task: develop RSA encryption/decryption and try breaking RSA ciphertexts



RSA cypher

- Algorithm consists of 2 main steps:
 - ✿ Key generation
 - ✿ Encryption/Decryption
- Pre-processing – converting string message to integer



RSA key generation

- Generate two different large primes p, q
- $n := p * q$
- $\phi(n) = (p - 1) * (q - 1)$ [Euler's totient]
- Choose random e between $\log_2(n)$ and $\phi(n)$ such that $\gcd(e, \phi(n)) = 1$
- Set $d := \text{inv}(e, \phi(n))$
- Public key is (e, n) and private key is (d, n)



Extended Euclidean algorithm

remainder[0] := n

remainder[1] := e

auxiliary[0] := 0

auxiliary[1] := 1

i := 2

while remainder[i] > 1

 remainder[i] := remainder(remainder[i-2] / remainder[i-1])

 quotient[i] := quotient(remainder[i-2] / remainder[i-1])

 auxiliary[i] := -quotient[i] * auxiliary[i-1] + auxiliary[i-2]

 i := i + 1

inverse := auxiliary[i]



Primality Test (Miller-Rabin)

write $n - 1$ as $2^s \cdot d$ with d odd

(by factoring powers of 2 from $n - 1$)

For i in $[0 .. k]$ (k - accuracy parameter)

$a := \text{random}(2, n - 2)$

$x := a^d \bmod n$

 if $x = 1$ or $x = n - 1$ then *continue*

 for $r = 1 .. s - 1$

$x := x^{2^r} \bmod n$

 if $x = 1$ then return ***composite***

 if $x = n - 1$ then *continue*

 return ***composite***

return ***probably prime***



RSA Encryption/Decryption

- Encryption : $c = m^e \bmod n$
- Decryption: $m = c^d \bmod n$
- Efficient calculation algorithm
 - ✳ Square-and-Multiply



Square-and-Multiply (x, c, n)

```
z := 1
for i := len - 1 downto 0
  z := pow(z, 2) mod n
  if bit(c, i) == 1 then
    z := (z * x) mod n
return z
```

len - number of bits in the binary representation of c
bit(c, i) - returns the i th bit in c



Attacking RSA cipher

- Fact: RSA is vulnerable for short m
- Idea of attack: Given the public key (e,n) a brute-force ciphertext-only attack may require to encrypt all possible m to see which one matches the ciphertext.



Attack algorithm

- Input:
 - ✿ Cipher text c
 - ✿ Public key (e, n)
 - ✿ k - the number of bits in plain text
- Calculate $i^e \bmod n$ for $i = [1.. 2^{(k/2)}]$. Keep track of which i gives which cipher text. Can use table or map for that.
- Loop over the table and calculate $x = c * \text{inv}(i^e, n) \bmod n$, try to find that x as j^e in the table
- If you can find j^e , then $m = i * j \bmod n$