Lecture 10: Attacking RSA

William Stein

Math 124 HARVARD UNIVERSITY **Fall 2001**

Nikita's public key is (n, e). If we compute the factorization of $n = pq$, then we can compute $\varphi(n)$ and hence deduce her secret decoding number d. Thus attempting to factor n is a way to try to break an RSA public-key cryptosystem. In this lecture we consider several approaches to "cracking" RSA, and relate them to the difficulty of factoring n.

1 Factoring n Given $\varphi(n)$

If you know $\varphi(n)$ then it is easy to factor n:
Suppose $n = pq$. Given $\varphi(n)$, it is very easy to compute p and q. We have

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1,$$

so we know both $pq = n$ and $p + q = n + 1 - \varphi(n)$. Thus we know the polynomial

$$x^2 - (p+q)x + pq = (x-p)(x-q)$$

whose roots are p and q. These roots can be found using the quadratic formula.

Example 1.1.

```
? n=nextprime(random(10^10))*nextprime(random(10^10));
? phin=eulerphi(n);
? f = x^2 - (n+1-phin)*x + n
%6 = x^2 - 12422732288*x + 31615577110997599711
? polroots(f)
%7 = [3572144239, 8850588049]
? n
%8 = 31615577110997599711
? 3572144239*8850588049
%9 = 31615577110997599711
```

2 When p and q Are Close

Suppose that p and q are "close" to each other. Then it is easy to factor n using a factorization method of Fermat.

Suppose $n = pq$ with $p > q$, say. Then

$$n = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2.$$

Since p and q are "close",

$$s = \frac{p-q}{2}$$

is small,

$$t = \frac{p+q}{2}$$

is only slightly larger than $\sqrt{n}$, and $t^2 - n = s^2$ is a perfect square. So we just try

$$t = \mathrm{ceil}(\sqrt{n}), \quad t = \mathrm{ceil}(\sqrt{n}) + 1, \quad t = \mathrm{ceil}(\sqrt{n}) + 2, \ldots$$

until $t^2 - n$ is a perfect square s^2. Then

$$p = t + s, \qquad q = t - s.$$

Example 2.1. Suppose $n = 23360947609$. Then

$$\sqrt{n} = 152842.88\ldots.$$

If $t = 152843$, then $\sqrt{t^2 - n} = 187.18\ldots$.
If $t = 152844$, then $\sqrt{t^2 - n} = 583.71\ldots$.
If $t = 152845$, then $\sqrt{t^2 - n} = 804 \in \mathbb{Z}$.
Thus $s = 804$. We find that $p = t + s = 153649$ and $q = t - s = 152041$.

Here is a bigger example in PARI:

```
? q=nextprime(random(10^50))
%20 = 78177096444230804504075122792410749354743712880803
? p=nextprime(q+1)  \\ a nearby prime
%21 = 78177096444230804504075122792410749354743712880899
? n=p*q
%22 = 61116584084505646970856342018459768505099085809499868895257O4...
      ...2596503421573992791632896516937224818 97
? t=floor(sqrt(n))+1
  ***   precision loss in truncation
? \p150        \\ set precision of floating-point computations.
   realprecision = 154 significant digits (150 digits displayed)
? t=floor(sqrt(n))+1
%29 = 78177096444230804504075122792410749354743712880851
? sqrt(t^2-n)
%30 = 48.000000000000000000000000000000000000000000000000000000000....
? s=48
%31 = 48
? t + s     \\ p
%33 = 78177096444230804504075122792410749354743712880899
? t - s     \\ q
%35 = 78177096444230804504075122792410749354743712880803
```

3 Factoring n Given d

Suppose that we crack an RSA cryptosystem by finding a d such that

$$a^{ed} \equiv a \pmod{n}$$

for all a. Then we've found an m $(= ed - 1)$ such that $a^m \equiv 1 \pmod{n}$ for all a with $\gcd(a, n) = 1$. Knowing a does not lead to a factorization of n in as direct a manner as knowing $\varphi(n)$ does (see Section 1). However, there is a probabilistic procedure that, given an m such that $a^m \equiv 1 \pmod{n}$, will with high probability find a factorization of n.

Probabilistic procedure to factor n:

1. m is even since $(-1)^m \equiv 1 \pmod{n}$.

2. If $a^{m/2} \equiv 1 \pmod{n}$ for all a coprime to n, replace m by $m/2$. In practice, it is not possible to determine whether or not this condition holds, because it would require doing a computation for too many a. Instead, we try a few random a; if $a^{m/2} \equiv 1 \pmod{n}$ for the a we check, then we divide m by 2. (If there exists even a single a such that $a^{m/2} \not\equiv 1 \pmod{n}$, then at least half the a have this property.)

   Keep repeating this step until we find an a such that $a^{m/2} \not\equiv 1 \pmod{n}$.

3. There is a 50% chance that a randomly chosen a will have the property that

   $$a^{m/2} \equiv +1 \pmod{p}, \qquad a^{m/2} \equiv -1 \pmod{q}$$

   or

   $$a^{m/2} \equiv -1 \pmod{p}, \qquad a^{m/2} \equiv +1 \pmod{q}.$$

   If the first case occurs, then

   $$p \mid a^{m/2} - 1, \qquad \text{but } q \nmid a^{m/2} - 1,$$

   so

   $$\gcd(a^{m/2} - 1, pq) = p,$$

   and we have factored n. Just keep trying a's until one of the cases occurs.

```
? \r rsa   \\ load the file rsa.gp, available at Lecture 9 web page.
? rsa = make_rsa_key(10)
%34 = [32295194023343, 29468811804857, 11127763319273]
? n = rsa[1]; e = rsa[2]; d = rsa[3];
? m = e*d-1
%38 = 327921963064646896263108960
? for(a=2,20, if(Mod(a,n)^m!=1,print(a)))   \\ prints nothing...
? m = m/2
%39 = 163960981532323448131554480
? for(a=2,20, if(Mod(a,n)^m!=1,print(a)))
```

```
? m = m/2
%40 = 8198049076616172406577724O
? for(a=2,20, if(Mod(a,n)^m!=1,print(a)))
? m = m/2
%41 = 4099024538308086203288862O
? for(a=2,20, if(Mod(a,n)^m!=1,print(a)))
? m = m/2
%42 = 2049512269154043101644431O
? for(a=2,20,if(Mod(a,n)^m!=1,print(a)))
2
5
6
... etc.
? gcd(2^m,n)
  ***   power overflow in pow_monome.
? x = lift(Mod(2,n)^m)-1
%43 = 4015382800098
? gcd(x,n)
%46 = 737531
? p = gcd(x,n)
%53 = 737531
? q = n/p
? p*q
%54 = 32295194023343
? n
%55 = 32295194023343
```

4 RSA Challenge n

The easiest challenge at

`http://www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html`

is the 576-bit number

```
Name:         RSA-576
Prize:        $10000
Digits:       174
Digit Sum:    785
188198812920607963838697239461650439807163563379417382700763356422988859
715234665485319060606504743045317388011303396716199692321205734031879550
656996221305168759307650257059
```