A variant of Wiener's attack on RSA

Andrej Dujella

Department of Mathematics
University of Zagreb, Croatia
e-mail: duje@math.hr
URL: http://web.math.hr/~duje/

RSA cryptosystem -
Rivest, Shamir, Adleman (1978)

$n = p \cdot q$, $\quad p$ and q are large primes

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

public exponent e, $\gcd(e, \varphi(n)) = 1$

secret exponent d, $ed \equiv 1 \pmod{\varphi(n)}$

In a typical RSA: p and q have approximately the same number of bits, and $e < n$.

encryption: $C = M^e \bmod n$

decryption: $M = C^d \bmod n$

To speed up the RSA decryption one may try to use small secret decryption exponent d. The choice of a small d is especially interesting when there is a large difference in computing power between two communicating devices, e.g. in communication between a smart card and a larger computer.

In this situation, it would be desirable:

smart card - small secret exponent

larger computer - small public exponent

to reduce the processing required in the smart card.

Wiener (1990) -
attack on RSA with small d:

$ed - k\varphi(n) = 1$

$\varphi(n) \approx n \quad \Rightarrow \quad \frac{k}{d} \approx \frac{e}{n}$

Assume that $p < q < 2p$. If $d < \frac{1}{3}n^{0.25}$, then

$$\left|\frac{k}{d} - \frac{e}{n}\right| < \frac{1}{2d^2}.$$

By classical Legendre's theorem, d is the denominator of some convergent p_m/q_m of the continued fraction expansion of e/n, and therefore d can be computed efficiently from the public key (n, e).

Total number of convergents is of order $O(\log n)$; a convergent can be tested in polynomial time.

Verheul and van Tilborg (1997): An extension of Wiener's attack that allows the RSA cryptosystem to be broken when d is a few bits longer than $n^{0.25}$. For $d > n^{0.25}$ their attack needs to do an exhaustive search for about $2t + 8$ bits (under reasonable assumptions on involved partial convergents), where $t = \log_2(d/n^{0.25})$.

Boneh and Durfee (1999),
Blömer and May (2001):
Attacks based on Coppersmith's lattice-based technique for finding small roots of modular polynomials equations using LLL-algorithm. The attacks works (heuristically, but practically) if $d < n^{0.292}$.

The conjecture is that the right bound below which a typical version of RSA is insecure is $d < n^{0.5}$.

D. (2004): A slight modification of the Verheul and van Tilborg attack, based on Worley's result from 1981 on Diophantine approximations, which implies that all rationals p/q satisfying the inequality

$$\left|\alpha - \frac{p}{q}\right| < \frac{c}{q^2},$$

for a positive real number c, are given by

$$\frac{p}{q} = \frac{rp_{m+1} \pm sp_m}{rq_{m+1} \pm sq_m}$$

for some $m \geq -1$ and nonnegative integers r and s such that $rs < 2c$.

D. and Ibrahimpašić (2008): Worley's result is sharp, in the sense that the condition $rs < 2c$ cannot be replaced by $rs < (2 - \varepsilon)c$ for any ε.

In both mentioned extensions of Wiener's attack, the candidates for the secret exponent are of the form $d = rq_{m+1} + sq_m$. We test all possibilities for d, and number of possibilities is roughly (number of possibilities for r) $\times$ (number of possibilities for s), which is $O(D^2)$, where $d = Dn^{0.25}$.

More precisely, number of possible pairs (r, s) in Verheul and van Tilborg attack is $O(D^2A^2)$, where $A = \max\{a_i : i = m+1, m+2, m+3\}$, while in our variant number of pairs is $O(D^2 \log A)$ (and also $O(D^2 \log D)$).

Another modification of the Verheul and van Tilborg attack has been recently proposed Sun, Wu an Chen. It requires (heuristically) an exhaustive search for about $2t - 10$ bits, so its complexity is also $O(D^2)$. We cannot expect drastic improvements here, since, by a result of Steinfeld, Contini, Wang and Pieprzyk from 2005, there does not exist an attack in this class with subexponential run-time.

Testing

There are two principal methods for testing:

1) compute p and q assuming d is correct guess:

$$\varphi(n) = (de-1)/k, \quad p+q = n+1-\varphi(n),$$

$$(p-q)^2 = (p+q)^2 - 4n;$$

2) test the congruence $(M^e)^d \equiv M \pmod{n}$, say for $M = 2$.

Here we present a new idea, which is to apply "meet-in-the-middle" to this second test.

We want to test whether
$2^{e(rq_{m+1}+sq_m)} \equiv 2 \pmod{n}$.
Note that m is (almost) fixed. Let m' be the largest odd integer such that

$$\frac{p_{m'}}{q_{m'}} > \frac{e}{n} + \frac{2.122e}{n\sqrt{n}}.$$

Then $m \in \{m', m'+1, m'+2\}$.

Let $2^{eq_{m+1}} \bmod n = a$, $(2^{eq_m})^{-1} \bmod n = b$. Then we test the congruence $a^r \equiv 2b^s \pmod{n}$.

We can do it by computing $a^r \bmod n$ for all r, sorting the list of results, and then computing $2b^s \bmod n$ for each s one at a time, and checking if the result appears in the sorted list.

This decrease the time complexity of testings phase to $O(D \log D)$ (with the space complexity $O(D)$).

We have implemented the proposed attack in PARI and C++ (with V. Petričević), and it works efficiently for values of D up to 2^{30}, i.e. for $d < 2^{30}n^{0.25}$.

For larger values of D the memory requirements become too demanded.

$\log_2 n$	$\log_2(2^{30}n^{0.25})$	$\log_2(n^{0.292})$
512	158	150
768	222	224
1024	286	299
2048	542	598

A space-time tradeoff might be possible, by using unsymmetrical variants of Worley's result (with different bounds on r and s).

bound for r	bound for s	chance of success
$4D$	$4D$	98%
$2D$	$2D$	89%
D	D	65%
D	$4D$	86%
$4D$	D	74%
$D/2$	$2D$	70%
$2D$	$D/2$	47%
$D/4$	$4D$	54%
$4D$	$D/4$	28%

The attack can be slightly improved by using better approximations to $\frac{k}{d}$, e.g. $\frac{e}{n+1-2\sqrt{n}}$ instead of $\frac{e}{n}$.

Implementation issues: hash functions instead of sorting.

With these improvements we hope that for 1024-bits RSA modulus n, the range in which our attack can be applied might be comparable with known attacks based on LLL-algorithm.