BREAKING RSA WITH DIOPHANTINE APPROXIMATION

ALEX YANG

Contents

Introduction

The RSA encryption algorithm, despite being introduced decades ago, remains secure by modern standards and is one of the most widely used encryption tools today. However, it must be used carefully, as many attacks have been developed to break its security when used improperly or when weak parameters are chosen.

The central theme of this paper is that Diophantine approximation, the study of approximating real numbers by rationals, provides a systematic way to turn "smallness" or "near divisibility" into cryptanalytic leverage against RSA. Many attacks on RSA reduce either to finding an unusually good rational approximation to some ratio determined by the public key, or to finding an unusually small integer root of a modular polynomial equation. Both of these are classical problems in Diophantine approximation, and so attacks that appear quite different at first share a common mathematical structure.

The paper is organized as follows. Section 1 reviews the RSA cryptosystem. Section 2 introduces continued fractions and presents Wiener's attack, which breaks RSA when the private key is small. Section 3 describes Coppersmith's method, a higher-dimensional generalization of the same approximation philosophy, and applies it to two further attacks: Håstad's broadcast theorem and a partial message exposure attack. These attacks are often presented separately; the goal here is to treat them together so that the role of Diophantine approximation in each becomes visible.

1. RSA

Suppose one would like to send a message to a friend over the internet while adversaries can intercept it with the goal of reading its contents. One of the best ways to ensure that adversaries cannot read the message is through RSA encryption, initially described by Rivest, Shamir, and Adleman in 1978 [5]. The encryption and decryption algorithm involves a public key known to everyone and a private key,

Date: April 2026.

known only to the recipient. After a public and private key have been generated, the sender can use the public key to encrypt the data and transmit it over the internet, after which the recipient decrypts it with the private key. Without the private key, adversaries are unable to decode the message. The encryption algorithm is presented below.

Key Generation. To generate keys, begin by selecting two large prime numbers p and q, roughly equal in magnitude. Then compute

$$n = pq$$

and Euler's totient function

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

Next, an integer e is chosen such that

$$1 < e < \varphi(n) \quad \text{and} \quad \gcd(e, \varphi(n)) = 1.$$

The integer e will be part of the public key.

Next, compute an integer d satisfying

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

so that d is the multiplicative inverse of e modulo $\varphi(n)$. Such a d exists precisely because $\gcd(e, \varphi(n)) = 1$, and it can be computed quickly via the extended Euclidean algorithm.

The public key is given by the pair (n, e), while the private key is (n, d).

Encryption. To send a message, the sender first encodes the message as an integer M such that

$$0 \leq M < n.$$

Using the recipient's public key (n, e), the sender computes

$$C \equiv M^e \pmod{n}.$$

The value C is the encrypted message.

Decryption. Upon receiving the ciphertext C, the recipient uses the private key d to compute

$$M \equiv C^d \pmod{n},$$

recovering the original message. To see why this is valid, note first that

$$C^d \equiv M^{ed} \pmod{n}.$$

Since $ed \equiv 1 \pmod{\varphi(n)}$, one can write $ed = 1 + k\varphi(n)$ for some integer k. By Euler's theorem, which states that $a^{\varphi(n)} \equiv 1 \pmod{n}$ for any a coprime to n,

$$M^{ed} = M^{1+k\varphi(n)} = M \cdot \left(M^{\varphi(n)}\right)^k \equiv M \pmod{n}$$

whenever $\gcd(M, n) = 1$. A slightly stronger argument using the Chinese remainder theorem shows that the identity $M^{ed} \equiv M \pmod{n}$ in fact holds for all M, not only those coprime to n.

Although RSA is considered secure, there are methods applicable to situations where poorly chosen private keys or small information leaks allow RSA's security to be broken. Many of these methods stem from the field of Diophantine approximation, which is introduced and applied in the following sections.

2. Continued Fractions and Wiener's Attack

Diophantine approximation is the study of approximating numbers by rationals. At its root is the question: given a number $x \in \mathbb{R}$, can one find $p, q \in \mathbb{Z}$ such that $x \approx \frac{p}{q}$? Dirichlet's approximation theorem [3] guarantees that for any irrational x, there are infinitely many pairs (p, q) such that

$$\left| x - \frac{p}{q} \right| < \frac{1}{q^2}.$$

The previous section left open the question of how an adversary might exploit the congruence $de \equiv 1 \pmod{\varphi(n)}$. One natural temptation is to choose a small private key d in order to save time decrypting RSA messages. This unfortunately leaves RSA vulnerable to Wiener's attack [6], which uses Diophantine approximation to break RSA schemes that have small private keys.

The motivation of the attack is the congruence

$$de \equiv 1 \pmod{\varphi(n)}.$$

One could of course try every possible d, hoping to guess the right one, but this would take $O(n)$ time, which is infeasible in typical RSA encryptions where n can be thousands of bits. Instead, the goal is to rewrite this congruence in a form where Diophantine approximation applies. There exists $k \in \mathbb{Z}$ such that

$$de - k\varphi(n) = 1, \tag{1}$$

and dividing through by $d\varphi(n)$ gives

$$\frac{e}{\varphi(n)} - \frac{k}{d} = \frac{1}{d\varphi(n)}.$$

Since $\varphi(n) = n - (p + q - 1)$ differs from n by an amount of order $\sqrt{n}$ in typical RSA implementations, the ratio $\frac{e}{n}$ is a close approximation of $\frac{k}{d}$. If there were an algorithm that efficiently found all the good rational approximations of a number, it could be run on $\frac{e}{n}$ in the hope of recovering $\frac{k}{d}$. This motivates the continued fractions algorithm, which does exactly that.

Continued Fractions Algorithm. For $x \in \mathbb{R}$, the continued fractions algorithm produces all $p, q \in \mathbb{Z}$ such that

$$\left| x - \frac{p}{q} \right| \leq \frac{1}{2q^2}.$$

In fact, the convergents of the continued fraction expansion are the "best possible" rational approximations to x in a precise sense: every rational $\frac{p}{q}$ satisfying the bound above must appear as a convergent. This is what makes the algorithm not merely convenient but mathematically essential for the attack below.

I will present the algorithm without proof of correctness. Given a real number x, define

$$x_0 = x.$$

For each $k \geq 0$, let

$$a_k = \lfloor x_k \rfloor.$$

If $x_k - a_k = 0$, the algorithm stops. Otherwise define

$$x_{k+1} = \frac{1}{x_k - a_k}.$$

This produces the continued fraction expansion

$$x = [a_0; a_1, a_2, a_3, \dots].$$

The associated convergents are defined recursively by

$$p_{-2} = 0, \quad p_{-1} = 1, \qquad q_{-2} = 1, \quad q_{-1} = 0,$$

and for $k \geq 0$,

$$p_k = a_k p_{k-1} + p_{k-2}, \qquad q_k = a_k q_{k-1} + q_{k-2}.$$

The rational numbers $\frac{p_k}{q_k}$ are called the convergents of the continued fraction.

To exhaust all approximations with denominator smaller than Q, this algorithm runs in $O(\log Q)$ time and produces $O(\log Q)$ convergents; the worst case scenario is approximating the golden ratio. With this tool in hand, it is now possible to use Wiener's attack, which breaks RSA when the private key is small.

Wiener's Attack. As usual, let (n, d) be the RSA private key, (n, e) the public key, and p, q primes such that $pq = n$. Wiener's attack is able to break RSA if

1. $$d < \frac{1}{3} n^{1/4}$$
2. $$p + q - 1 < 3\sqrt{n}.$$

The first condition states that d is relatively small. The second condition may seem strange at first, but in typical RSA implementations p and q are both roughly $\sqrt{n}$ to maximize security; in particular, they are usually of the same bit length, which makes this condition true.

The attack proceeds as follows:

(1) Compute the continued fraction expansion of $\frac{e}{n}$.
(2) Compute its convergents $\frac{k_i}{d_i}$.
(3) For each convergent, test whether

$$\varphi(n) = \frac{ed_i - 1}{k_i}$$

is an integer.
(4) If so, compute

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

Then solve the quadratic

$$x^2 - (p + q)x + n = 0.$$

If this yields integer roots p and q with $pq = n$, then d_i is the private exponent.

Proof. The strategy is to show that the hypotheses of the attack force $\frac{k}{d}$ to be a sufficiently good rational approximation to $\frac{e}{n}$ that the continued fractions algorithm must output it.

From the assumption $p+q-1<3\sqrt{n}$,

$$\varphi(n)=(p-1)(q-1)=pq-p-q+1>n-3\sqrt{n},$$

so $n-\varphi(n)<3\sqrt{n}$. Recall from (1) that $ed-\varphi(n)k=1$ for some $k\in\mathbb{Z}$. Then

$$\begin{aligned}
\frac{e}{n}-\frac{k}{d} &= \frac{ed-nk}{nd} && \text{(common denominator)},\\
&= \frac{ed-\varphi(n)k-nk+\varphi(n)k}{nd} && \text{(adding and subtracting } \varphi(n)k),\\
&= \frac{1-k(n-\varphi(n))}{nd} && \text{(by (1))},\\
&> \frac{1-3k\sqrt{n}}{nd} && \text{(since } n-\varphi(n)<3\sqrt{n}).
\end{aligned}$$

Both sides are negative (since $k(n-\varphi(n))\gg 1$ and $3k\sqrt{n}\gg 1$), so taking absolute values reverses the inequality and gives

$$\left|\frac{e}{n}-\frac{k}{d}\right|<\left|\frac{1-3k\sqrt{n}}{nd}\right|<\frac{3k\sqrt{n}}{nd}=\frac{3k}{d\sqrt{n}}.$$

Since $k\varphi(n)<ed$ and $e<\varphi(n)$, one has $k<d<\frac{1}{3}n^{1/4}$. Therefore,

$$\left|\frac{e}{n}-\frac{k}{d}\right|\leq\frac{1}{dn^{1/4}}<\frac{1}{2d^2}.$$

This is precisely the condition needed for $\frac{k}{d}$ to appear among the convergents produced by the continued fractions algorithm applied to $\frac{e}{n}$. Running that algorithm therefore yields the correct pair (k,d) among $O(\log n)$ candidates, each of which can be checked directly against the public data. □

3. The Coppersmith Method

From Diophantine Approximation to Coppersmith. Wiener's attack exploited the congruence $de-k\varphi(n)=1$ by reinterpreting it as a rational-approximation problem. The same philosophy, turning a cryptographic congruence into an approximation problem, can be pushed much further, but doing so requires replacing rationals with polynomials and continued fractions with lattice reduction.

In classical Diophantine approximation, one aims to find integers p,q such that

$$\left|\alpha-\frac{p}{q}\right|$$

is small. In Wiener's attack, this idea was used to approximate $\frac{e}{n}$ by $\frac{k}{d}$.

Coppersmith's method extends this idea to polynomial equations. Instead of approximating real numbers, one seeks small integer solutions to equations of the form

$$f(x)\equiv 0 \pmod{n}.$$

Rewriting this as

$$f(x)=kn,$$

one sees that the problem is again to find a small value of x at which $f(x)$ is exactly divisible by n. Thus, Coppersmith's method can be viewed as a higher-dimensional generalization of Diophantine approximation, where lattice reduction replaces continued fractions. The added flexibility gives rise to several further attacks on RSA.

In Wiener's attack, the polynomial was essentially forced on the attacker: it had to come from the congruence $ed - 1 \equiv 0 \pmod{n}$. With Coppersmith's method, the attacker has greater flexibility in choosing the polynomial on the left-hand side, which enables several more creative attacks [2].

Coppersmith Method. Let $n \in \mathbb{Z}$, and let $f(x) \in \mathbb{Z}[x]$ be a monic polynomial of degree d. Then there exists a polynomial-time algorithm that finds all integers x_0 satisfying

$$f(x_0) \equiv 0 \pmod{n} \quad \text{with} \quad |x_0| < n^{1/d-\epsilon},$$

for fixed $\epsilon > 0$. The running time is polynomial in $\frac{1}{\epsilon}$ and $\log n$. In most applications, the exponent of n will not be extremely close to $\frac{1}{d}$, so the $\frac{1}{\epsilon}$ factor in the time complexity is negligible.

The method uses the LLL algorithm, which finds a basis of a lattice whose vectors are both short and nearly orthogonal. Here "nearly orthogonal" means that each basis vector has a small projection onto the span of the previous ones, so that the pairwise inner products $\langle v_i, v_j \rangle$ are small relative to the vectors' norms.

Proof Sketch of Coppersmith's Theorem. A full proof is omitted here since it needs more lattice theory than has been built up, but the core idea is worth sketching.

Suppose the goal is to find x_0 with $f(x_0) \equiv 0 \pmod{n}$ and $|x_0| < X$. The strategy is to build many new polynomials that also vanish at x_0 modulo large powers of n, and then combine them into a single polynomial h with coefficients small enough that $h(x_0) = 0$ holds over the integers, not merely modulo n.

Concretely, one builds polynomials of the form

$$g_{i,j}(x) = x^j n^i f(x)^{m-i}.$$

Since $n \mid f(x_0)$, each $g_{i,j}$ vanishes at x_0 modulo n^m. One then substitutes $x \mapsto Xx$ — a rescaling that encodes the size bound $|x_0| < X$ into the coefficient vectors, so that short vectors in the resulting lattice correspond to polynomials with small values at x_0 — and stacks these coefficient vectors into a lattice. Any short vector in this lattice corresponds to a polynomial h that still makes $h(x_0)$ small modulo n^m.

If h has small coefficients and x_0 is small, then $|h(x_0)|$ is small too. This forces $h(x_0) = 0$, since $h(x_0)$ must also be divisible by n^m. At that point x_0 is just a root of an integer polynomial, and it can be found by standard methods.

The LLL algorithm is what finds the short lattice vector efficiently. The bound $X < n^{1/d-\epsilon}$ comes from ensuring that $h(x_0)$ is actually forced to zero, which requires the lattice to be set up carefully.

Repeated Message Attacks. Coppersmith's method is powerful because the attacker is free to choose the polynomial f. One natural setting where this freedom pays off is when the same message is broadcast to multiple recipients, possibly after being transformed by some public padding scheme.

Consider the case where a sender wishes to broadcast a private message to more than one friend. It can be shown that if the message is sent to at least three friends under a common low public exponent (e.g., $e = 3$), an adversary can reveal the contents of the message without any of the friends' private keys. A natural countermeasure is to pad the message differently for each recipient, or to add a recipient-dependent constant before encryption. However, this additional security is not foolproof: as long as the adversary knows the padding or constant-adding function, the transformation is a polynomial function of the message, and Coppersmith's method can be brought to bear.

Håstad's broadcast theorem uses the Coppersmith method to break repeated messages that have been transformed in any fixed polynomial manner [4].

Håstad's Broadcast Theorem. Let $N_1, \ldots, N_k$ be pairwise relatively prime positive integers, and define

$$N_{\min} = \min_i N_i.$$

Let $g_i(x) \in (\mathbb{Z}/N_i\mathbb{Z})[x]$ be polynomials of maximum degree d. Suppose there exists a unique integer $M < N_{\min}$ such that

$$g_i(M) \equiv 0 \pmod{N_i} \quad \text{for all } i = 1, \ldots, k.$$

If $k > d$, then M can be efficiently recovered from the data $\{(N_i, g_i)\}_{i=1}^k$.

To see how this applies to RSA, suppose the sender uses padding by appending 123 to the end of the message M. The padded message becomes $1000M + 123$, and the ciphertext sent to recipient i is $C_i \equiv (1000M + 123)^e \pmod{N_i}$. The polynomial $g_i(x) = (1000x + 123)^e - C_i$ then satisfies $g_i(M) \equiv 0 \pmod{N_i}$, and the hypotheses of the theorem apply once at least $d = e$ such equations have been intercepted.

Proof. Let

$$\mathcal{N} = N_1 N_2 \cdots N_k.$$

One may assume that each $g_i(x)$ is monic. If not, then the leading coefficient would fail to be invertible modulo N_i, which would reveal a nontrivial factor of N_i and break the ith RSA system outright. By multiplying by appropriate powers of x, one may also assume that each g_i has degree exactly d.

Using the Chinese remainder theorem, construct integers T_i such that

$$T_i \equiv 1 \pmod{N_i}, \quad \text{and} \quad T_i \equiv 0 \pmod{N_j} \text{ for } j \neq i.$$

Define the polynomial

$$g(x) = \sum_{i=1}^{k} T_i g_i(x).$$

The leading coefficient of g is $\sum_{i=1}^k T_i$, since each g_i is monic of degree d. For each fixed i, the terms T_j with $j \neq i$ are all $0 \pmod{N_i}$ while $T_i \equiv 1 \pmod{N_i}$, so $\sum_j T_j \equiv 1 \pmod{N_i}$. Since this holds for every i and the N_i are pairwise coprime, the Chinese remainder theorem gives $\sum_j T_j \equiv 1 \pmod{\mathcal{N}}$. Working in $\mathbb{Z}/\mathcal{N}\mathbb{Z}$, one may therefore treat g as monic.

Furthermore,

$$g(M) \equiv 0 \pmod{\mathcal{N}},$$

because each $g_i(M) \equiv 0 \pmod{N_i}$. Since $M < N_{\min} \leq \mathcal{N}^{1/k}$ and $k > d$,

$$M < \mathcal{N}^{1/d}.$$

By Coppersmith's theorem, M can be recovered efficiently. □

In particular, if all $e_i = e$ and the messages are linearly related, then the plaintext can be recovered as soon as $k > e$.

Defending against Håstad's Broadcast Theorem. To prevent this attack, randomized padding is necessary. Any deterministic or structured padding scheme is potentially vulnerable as long as the adversary can model it as a polynomial.

Partial Information Attacks. Coppersmith's method also applies in a rather different setting: that of partial plaintext knowledge. Suppose, for instance, that an adversary learns that every message begins with a fixed phrase such as a standard greeting. If enough of the message is known, that information alone can suffice to break the encryption [1].

Partial Message Exposure Attack. Suppose Bob's public key is (N, e), and Alice encrypts a message M by computing

$$C \equiv M^e \pmod{N}.$$

Assume that an attacker knows part of the message M, and only a small portion remains unknown. More precisely, suppose

$$M = M_0 + x,$$

where M_0 is known and x is an unknown integer satisfying

$$|x| < X$$

for some sufficiently small bound X. The ciphertext equation becomes

$$C \equiv (M_0 + x)^e \pmod{N},$$

or equivalently

$$f(x) = (M_0 + x)^e - C \equiv 0 \pmod{N}.$$

Recovering the unknown portion of the message is therefore equivalent to finding a small root x of the polynomial $f(x)$ modulo N. Since $f(x)$ is a monic polynomial of degree e, Coppersmith's theorem implies that all such roots can be found efficiently whenever

$$|x| < N^{1/e}.$$

Heuristically, this bound translates into a statement about bit length: if N is an n-bit number and the public exponent is e, then Coppersmith's method recovers the message as long as the unknown portion has fewer than approximately $\frac{n}{e}$ bits. This is not a sharp theorem statement (the ϵ in Coppersmith's theorem and implementation considerations introduce small losses), but it gives the right order of magnitude and is the usual rule of thumb.

For large e, an adversary must therefore know a substantial part of the message in order to deduce the rest. In the past, $e = 3$ was a common public exponent, as it makes encryption extremely fast and is still mostly secure against naive attacks. However, small exponents leave RSA vulnerable to precisely the low-exponent attacks discussed above. The default public exponent is now $65537 = 2^{16} + 1$, which strikes a balance between two competing concerns: it is large enough that the $N^{1/e}$ bound in Coppersmith's theorem leaves only a tiny window for partial-message attacks, while its binary form $(10000000000000001)_2$ means that modular exponentiation still takes very few multiplications and so remains very fast [1].

References

[1] Dan Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society*, 46(2):203–213, 1999.

[2] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[3] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 6 edition, 2008.

[4] Johan Håstad. On using RSA with low exponent in a public key network. In Hugh C. Williams, editor, *Advances in Cryptology—CRYPTO '85 Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 403–408. Springer, Berlin, Heidelberg, 1986.

[5] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[6] Michael J. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, 1990.