The RSA cryptosystem

Part 2: attacks against RSA

Jean-Sébastien Coron

University of Luxembourg

The RSA cryptosystem

- RSA key generation:
  - Generate two large distinct primes p and q of same bit-size $k/2$, where k is a parameter.
  - Compute $N = p \cdot q$ and $\phi = (p-1)(q-1)$.
  - Select a random integer e, $1 < e < \phi$ such that $\gcd(e, \phi) = 1$
  - Compute the unique integer d such that

    $$e \cdot d \equiv 1 \pmod{\phi}$$

    using the extended Euclidean algorithm.
  - The public key is (N, e).
  - The private key is d.

Textbook RSA encryption

- Encryption
  - Given a message $m \in [0, N-1]$ and the recipent's public-key (n, e), compute the ciphertext:

$$c = m^e \bmod N$$

- Decryption
  - Given a ciphertext c, to recover m, compute:

$$m = c^d \bmod N$$

- Textbook RSA encryption is insecure
  - One must first apply a probabilistic encoding to m
  - Encryption: $c = \mu(m, r)^e \bmod N$
  - Decryption: compute $c^d \bmod N$, check that the encoding is correct, and recover m
  - Example: OAEP

Attacks against RSA

- Mathematical attacks against RSA
  - Factoring. Elementary attacks against textbook RSA encryption and signature. Previous lecture.
  - Low private / public exponent attacks. Coppersmith's technique. This lecture.
  - Attacks against RSA signatures. Next lecture.
- Implementation attacks
  - Timing attacks, power attacks and fault attacks
  - Countermeasures

Low private exponent attacks

- To reduce decryption time, one could use a small d
  - $m = c^d \bmod N$
  - Decryption time is proportional to the bitsize of d
  - First generate a small d, and compute the (full-size) e such that $e \cdot d = 1 \pmod{\phi(N)}$
- Wiener's attack
  - recover d if $d < N^{0.25}$
  - based on rational reconstruction

Rational reconstruction

- Rational reconstruction
  - Given u, e such that $a \cdot u \equiv b \pmod{e}$ with $2|a| \cdot |b| < e$, recover the integers a and b.
- Can be solved by modifying the extended Euclidean algorithm
  - The extended Euclidean algorithm computes a sequence a_i, b_i such that $a_i \cdot u \equiv b_i \pmod{e}$, where a_i is increasing, and b_i is decreasing.
  - Stop when $|a_i| \leq A$ and $|b_i| \leq B$ for upper-bounds A, B with $2A \cdot B < e$

Wiener's attack on small d

- We have $d \cdot e = 1 + a \cdot \phi(N)$ for some $a \in \mathbb{Z}$
  - With $\phi(N) = (p-1)(q-1) = N - x$, this gives:

$$a \cdot (N - x) \equiv -1 \pmod{e}$$

- This gives $a \cdot N \equiv u \pmod{e}$ with $u = ax - 1$
  - If $d \simeq N^{1/4}$, then $a \simeq N^{1/4}$ and $u \simeq N^{3/4}$.
  - Since $|a| \cdot |u| \simeq N \simeq e$, we can recover a and u by rational reconstruction.
- From a and u, we recover x. From x we recover $\phi(N)$. From e and $\phi(N)$ we recover the private exponent d.

Extension of Wiener's attack

- Wiener's attack
  - recover d if $d < N^{0.25}$
- Boneh and Durfee's attack (1999)
  - Recover d if $d < N^{0.29}$
  - Based on lattice reduction and Coppersmith's technique
  - Open problem: extend to $d < N^{0.5}$
- Conclusion: devastating attack
  - Use a full-size d

Low public exponent attack

- To reduce encryption time, one can use a small e
  - $c = m^e \mod N$
  - For example $e = 3$ or $e = 2^{16} + 1$
- Coppersmith's theorem :
  - Let N be an integer and f be a polynomial of degree δ. Given N and f, one can recover in polynomial time all x_0 such that $f(x_0) = 0 \pmod{N}$ and $|x_0| < N^{1/\delta}$.
- Application: partially known message attack :
  - If $c = (B\|m)^3 \mod N$, one can recover m if $\mathrm{sz}(m) < \mathrm{sz}(N)/3$
  - Define $f(x) = (B \cdot 2^k + x)^3 - c \pmod{N}$.
  - Then $f(m) = 0 \pmod{N}$ and apply Coppersmith's theorem to recover m.

Coppersmith's theorem for solving modular polynomial equations

- Solving $f(x) = 0 \pmod{N}$ when N is of unknown factorization: hard problem.
  - For $f(x) = x^2 - a$, equivalent to factoring N.
  - For $f(x) = x^e - a$, equivalent to inverting RSA.
- Coppersmith showed (E96) that finding small roots is easy.
  - When $\deg f = \delta$, finds in polynomial time all integer x_0 such that $f(x_0) = 0 \pmod{N}$ and $|x_0| \leq N^{1/\delta}$.
  - Based on the LLL lattice reduction algorithm.

Coppersmith's bound

- Coppersmith's theorem
  - When $\deg f = \delta$, finds in polynomial time all integer x_0 such that $f(x_0) \equiv 0 \pmod{N}$ and $|x_0| \leq N^{1/\delta}$.
- Consider the particular case $f(x) = x^\delta - a$
  - We want to solve $f(x_0) = 0 \pmod{N}$ with $|x_0|^\delta < N$
  - This gives $(x_0)^\delta \equiv a \pmod{N}$ with $|x_0|^\delta < N$
  - This implies $(x_0)^\delta = a$ over $\mathbb{Z}$
  - $x_0 = a^{1/\delta}$ over $\mathbb{Z}$
- Coppersmith's theorem is a generalization to any polynomial $f(x)$ modulo N of degree δ,
  with the same bound.

Applications in cryptanalysis

- Coppersmith's technique for finding small roots of polynomial equations [Cop97]
  - Based on the LLL lattice reduction algorithm
- Numerous applications in cryptanalysis :
  - Partially known message attack with $c = (B \| m)^3 \pmod{N}$
  - Coppersmith's short pad attack with $c_1 = (m \| r_1)^3 \pmod{N}$ and $c_2 = (m \| r_2)^3 \pmod{N}$
  - Factoring $N = pq$ when half of the bits of p are known
  - Factoring $N = p^r q$ for large r (Boneh et al., C99).

Solving $x^2 + ax + b = 0 \pmod{N}$

- Illustration with a polynomial of degree 2 :
  - Let $f(x) = x^2 + ax + b \pmod{N}$.
  - We must find x_0 such that $f(x_0) = 0 \pmod{N}$ and $|x_0| \le X$.
- We are interested in finding a small linear integer combination of the polynomials $f(x)$, Nx and N:
  - $h(x) = \alpha \cdot f(x) + \beta \cdot Nx + \gamma \cdot N$
  - Then $h(x_0) = 0 \pmod{N}$.
- If the coefficients of $h(x)$ are small enough :
  - Since x_0 is small, $h(x_0)$ will be small. If $|h(x_0)| < N$, then $h(x_0) = 0 \pmod{N} \Rightarrow h(x_0) = 0$ over $\mathbb{Z}$.
  - We can recover x_0 using any root-finding algorithm.

Solving $x^2 + ax + b = 0 \pmod{N}$

- From $h(x) = \alpha \cdot f(x) + \beta \cdot Nx + \gamma \cdot N$
  - with $f(x) = x^2 + ax + b$
  - we get $h(x) = \alpha x^2 + (\alpha \cdot a + \beta \cdot N)x + \alpha \cdot b + \gamma \cdot N$
- We want $|h(x_0)| < N$
  - True if $|\alpha x_0^2| < N/3$ and $|\alpha \cdot a + \beta \cdot N| \cdot |x_0| < N/3$ and $|\alpha \cdot b + \gamma \cdot N| < N/3$
  - With $|x_0| < X$, true if $|\alpha X^2| < N/3$ and $|\alpha \cdot a + \beta \cdot N| \cdot X < N/3$ and $|\alpha \cdot b + \gamma \cdot N| < N/3$
- True if $\|\alpha[X^2, aX, b] + \beta[0, NX, 0] + \gamma[0, 0, N]\| < N/3$
  - How do we find such integers α, β, γ ?
  - With the LLL lattice reduction algorithm.

Using LLL lattice reduction

- We want $\|\alpha[X^2, aX, b] + \beta[0, NX, 0] + \gamma[0, 0, N]\| < N/3$
  - Let L be the corresponding lattice, with a basis of row vectors :
$$L = \begin{bmatrix} X^2 & aX & b \\ & NX & \\ & & N \end{bmatrix}$$
  - Using LLL, one can find a lattice vector $\vec{b}$ of norm :
$$\|\vec{b}\| \leq 2(\det L)^{1/3} = 2N^{2/3}X$$
  - $\vec{b} = \alpha[X^2, aX, b] + \beta[0, NX, 0] + \gamma[0, 0, N]$
- We want $\|\vec{b}\| < N/3$
  - True if $2N^{2/3}X < N/3$
  - True if $X < N^{1/3}/6$
  - We recover x_0 by finding the roots over $\mathbb{Z}$ of $h(x) = \alpha f(x) + \beta Nx + \gamma$

Sage code

```
"Finds a small root of polynomial x^2+ax+b=0 mod N"
def CopPolyDeg2(a,b,Nn):
  n=Nn.nbits()
  X=2^(n//3-3)
  M=matrix(ZZ,[[X^2,a*X,b],\
               [0   ,Nn*X,0],\
               [0   ,0   ,Nn]])
  V=M.LLL()
  v=V[0]
  R.<x> = ZZ[]
  h=sum(v[i]*x^(2-i)/X^(2-i) for i in range(3))
  return h.roots()
```

Lattices and lattice reduction

- Definition :
  - Let $\vec{u}_1, \ldots, \vec{u}_\omega \in \mathbb{Z}^n$ be linearly independent vectors with $\omega \le n$. The lattice L spanned by the $\vec{u}_i$'s is
$$L = \left\{ \sum_{i=1}^{\omega} \alpha_i \cdot \vec{u}_i \mid \alpha_i \in \mathbb{Z} \right\}$$
  - If L is full rank ($\omega = n$), then $\det L = |\det M|$, where M is the matrix whose rows are the basis vectors $\vec{u}_1, \ldots, \vec{u}_\omega$.
- The LLL algorithm :
  - The LLL algorithm, given $(\vec{u}_1, \ldots, \vec{u}_\omega)$, finds in polynomial time a vector $\vec{b}_1$ such that:
$$\|\vec{b}_1\| \le 2^{(\omega-1)/4} \det(L)^{1/\omega}$$

Improving the bound on $|x_0|$

- The previous bound gives $|x_0| \leq N^{1/3}/6$ for a polynomial of degree 2
    - But Coppersmith's bound gives $|x_0| \leq N^{1/2}$.
- Technique : work modulo N^ℓ instead of N.
    - Example with $\ell = 2$:
    - Let $g(x) = f(x)^2$. Then $g(x_0) = 0 \pmod{N^2}$.
    - $g(x) = x^4 + a'x^3 + b'x^2 + c'x + d'$.
    - Find a small linear combination $h(x)$ of the polynomials $g(x)$, $Nxf(x)$, $Nf(x)$, N^2x and N^2.
    - Then $h(x_0) = 0 \pmod{N^2}$.
    - If the coefficients of $h(x)$ are small enough, then $h(x_0) = 0$.

Details when working modulo N^2

- Lattice basis with the coefficients of the polynomials $g(xX)$, $NxXf(xX)$, $Nf(xX)$, N^2xX and N^2.

$$\begin{bmatrix} X^4 & a'X^3 & b'X^2 & c'X & d' \\ & NX^3 & NaX^2 & NbX & \\ & & NX^2 & NaX & Nb \\ & & & N^2X & \\ & & & & N^2 \end{bmatrix} \quad \begin{matrix} g(x) \\ Nxf(x) \\ Nf(x) \\ N^2x \\ N^2 \end{matrix}$$

- Using LLL, one gets a polynomial $h(xX)$ with:
  - $\|h(xX)\| \leq 2 \cdot (\det L)^{1/5} \leq 2X^2N^{6/5}$
  - If $X < N^{2/5}/4$, then $\|h(xX)\| < N^2/5$ and we must have $h(x_0) = 0$.
  - Improved bound $N^{2/5}$ instead of $N^{1/3}$.

Coppersmith's algorithm for finding the small roots of $f(x) = 0 \pmod N$

- Find a small linear integer combination $h(x)$ of the polynomials :
  - $q_{ik}(x) = x^i \cdot N^{\ell-k} f^k(x) \pmod{N^\ell}$
  - For some ℓ and $0 \le i < \delta$ and $0 \le k \le \ell$.
  - $f(x_0) = 0 \pmod N \Rightarrow f^k(x_0) = 0 \pmod{N^k} \Rightarrow q_{ik}(x_0) = 0 \pmod{N^\ell}$.
  - Then $h(x_0) = 0 \pmod{N^\ell}$.
- If the coefficients of $h(x)$ are small enough :
  - Then $h(x_0) = 0$ holds over $\mathbb{Z}$.
  - x_0 can be found using any standard root-finding algorithm.
- For large enough ℓ, recovers all roots $|x_0| < N^{1/\delta}$ of $f(x) = 0 \pmod N$ where $\delta = \deg f$.

Another low public exponent attack

- Coppersmith's short pad attack
  - Let $c_1 = (m \| r_1)^3 \pmod N$ and $c_2 = (m \| r_2)^3 \pmod N$
  - One can recover m if $r_1, r_2 < N^{1/9}$
  - Let $g_1(x, y) = x^3 - c_1$ and $g_2(x, y) = (x + y)^3 - c_2$.
  - g_1 and g_2 have a common root $(m \| r_1, r_2 - r_1)$ modulo N.
  - $h(y) = \mathrm{Res}_x(g_1, g_2)$ has a root $\Delta = r_2 - r_1$, with $\deg h = 9$.
  - To recover $m \| r_1$, take gcd of $g_1(x, \Delta)$ and $g_2(x, \Delta)$.
- Conclusion:
  - Attack only works for specific encryption schemes.
  - Low public exponent is secure when provably secure construction is used,
    for example OAEP.

Factoring with high bits known

- Let $N = p \cdot q$. Assume that we know half of the most significant bits of p.
  - Write $p = P + x_0$ for some known P and unknown x_0 with $x_0 < p^{1/2}$.
- Consider the system:

$$\begin{cases} N \equiv 0 \pmod{P + x_0} \\ x + P \equiv 0 \pmod{P + x_0} \end{cases}$$

  - x_0 is a small root of both polynomial equations.
  - Apply Coppersmith's technique with unknown modulus $P + x_0$.
  - We can recover x_0 if $x_0 < p^{1/2}$
- Polynomial time factorization of $N = pq$
  if half of the high order (or low order)
  bits of p are known.

Example of factoring with high bits known

- Let $N = pq$ with $p = P + x_0$ for known P and $|x_0| < X$
- Consider the lattice of row vectors:

$$L = \begin{bmatrix} X^2 & PX & \\ & X & P \\ & & N \end{bmatrix} \qquad \begin{matrix} x^2 + Px \\ x + P \\ N \end{matrix}$$

- A short vector $\vec{b} \in L$ gives a polynomial $h(x)$ such that
  - $h(x) = \alpha(x + P)x + \beta(x + P) + \gamma N$
  - $h(x_0) \equiv 0 \pmod{P + x_0}$ because $N \equiv 0 \pmod{P + x_0}$
  - If $|h(x_0)| < P + x_0$, then $h(x_0) = 0$ and we can recover x_0

Analysis

$$L = \begin{bmatrix} X^2 & PX & \\ & X & P \\ & & N \end{bmatrix}$$

- With LLL, we obtain $\|\vec{b}\| \leq 2\det^{1/3} L = 2XN^{1/3}$
  - $h(x) = \alpha(x+P)x + \beta(x+P) + \gamma N$
  - We have $|h(x_0)| \leq 3\|\vec{b}\| \leq 6XN^{1/3}$
  - We want $|h(x_0)| < P + x_0 = p$.
  - We know $N^{1/2}/2 < p$ when $2^{k/2-1} < p, q < 2^{k/2}$
  - True if $6XN^{1/3} < N^{1/2}/2$. This gives $X < N^{1/6}/12$
- We can recover the factorization of $N = pq$ if we know $2/3$ of the high-order bits of p
  - We can reach $1/2$ with higher dimensional matrices

Factoring $N = p^r q$ in Polynomial Time

- Extension to $N = p^r q$ from [BDHG99]
  - Polynomial-time factorization of $N = p^r q$ when $1/(r+1)$ of the bits of p are known.
- Polynomial-time factorization of $N = p^r q$ for large r
  - When $r \simeq \log p$, only a constant number of bits of p need to be known.
  - Exhaustive search of these bits is then polynomial-time
- In practice, unpractical compared to the (subexponential) Elliptic Curve factoring Method (ECM).

Applications of Coppersmith's technique

- Coppersmith's technique for finding small roots of polynomial equations [Cop97]
  - Based on the LLL lattice reduction algorithm
- Numerous applications in cryptanalysis :
  - Partially known message attack with $c = (B\|m)^3 \pmod{N}$
  - Coppersmith's short pad attack with $c_i = (m\|r_i)^3 \pmod{N}$
  - Factoring $N = pq$ with high bits known [Cop97]
  - Factoring $N = p^r q$ for large r [BDHG99]
  - Breaking RSA for $d < N^{0.29}$ [BD99]
- Other applications
  - Cryptanalysis of RSA with small CRT exponents [JM07]
  - Deterministic equivalence between recovering d and factoring N [May04]
  - Improved security proof for RSA-OAEP with low public exponent (Shoup, C01).

Appendix

Howgrave-Graham lemma

- Given $h(x) = \sum h_i x^i$, let $\|h\|^2 = \sum h_i^2$.
- Howgrave-Graham lemma :
  - Let $h \in \mathbb{Z}[x]$ be a sum of at most ω monomials. If $h(x_0) = 0 \pmod{N}$ with $|x_0| \leq X$ and $\|h(xX)\| < N/\sqrt{\omega}$, then $h(x_0) = 0$ holds over $\mathbb{Z}$.
  - Proof :

$$\begin{aligned} |h(x_0)| &= \left|\sum h_i x_0^i\right| = \left|\sum h_i X^i \left(\frac{x_0}{X}\right)^i\right| \\ &\leq \sum \left|h_i X^i \left(\frac{x_0}{X}\right)^i\right| \leq \sum |h_i X^i| \\ &\leq \sqrt{\omega}\|h(xX)\| < N \end{aligned}$$

Since $h(x_0) = 0 \mod N$,
this gives $h(x_0) = 0$.