Recent Attacks on the RSA Cryptosystem

Abderrahmane Nitaj

University of Caen Basse Normandie, France

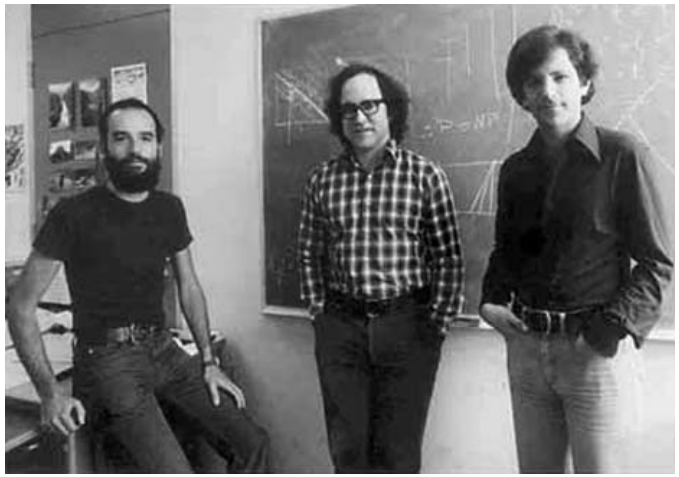
Putra Jaya, Malaysia, June 24, 2014

Contents

Contents

The RSA Cryptosystem

- Invented in 1978 by Rivest, Shamir and Adleman.



- The most widely used asymmetric cryptosystem.
- The security of RSA is based on two hard problems:
  1. The integer factorization problem.
  2. The RSA Problem (the eth modular root).

The RSA Cryptosystem

The most widely used cryptosystem

1. Encryption and digital signature.
2. Implemented in most Web servers and browsers.
3. Securing e-commerce and e-mail.
4. Authenticity of electronic documents.
5. Most commercially available security products.
6. Alleged NSA backdoor in random number generator proposed and used by RSA .

The RSA Cryptosystem

Key Generation

1. Generate two large primes p and q of the same bit size.
2. Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$.
3. Choose a random e with $1 < e < \phi(N)$ such that $\gcd(e, \phi(N)) = 1$.
4. Compute $d \equiv e^{-1} \pmod{\phi(N)}$.
5. Publish the public key (N, e).
6. The private key is (N, d).

Encryption

1. Compute $c \equiv m^e \pmod{N}$.
2. Send the ciphertext c.

Decryption

1. Compute $m \equiv c^d \pmod{N}$.

The RSA Cryptosystem

Key Generation

1. Generate two large primes p and q of the same bit size.
2. Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$.
3. Choose a random e with $1 < e < \phi(N)$ such that $\gcd(e, \phi(N)) = 1$.
4. Compute $d \equiv e^{-1} \pmod{\phi(N)}$.
5. Publish the public key (N, e).
6. The private key is (N, d).

Encryption

1. Compute $c \equiv m^e \pmod{N}$.
2. Send the ciphertext c.

Decryption

1. Compute $m \equiv c^d \pmod{N}$.

The RSA Cryptosystem

Key Generation

1. Generate two large primes p and q of the same bit size.
2. Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$.
3. Choose a random e with $1 < e < \phi(N)$ such that $\gcd(e, \phi(N)) = 1$.
4. Compute $d \equiv e^{-1} \pmod{\phi(N)}$.
5. Publish the public key (N, e).
6. The private key is (N, d).

Encryption

1. Compute $c \equiv m^e \pmod{N}$.
2. Send the ciphertext c.

Decryption

1. Compute $m \equiv c^d \pmod{N}$.

RSA: The hard problems

The equations

$$N = pq, \qquad \phi(N) = (p-1)(q-1),$$
$$ed - k\phi(N) = 1, \quad c \equiv m^e \pmod{N}.$$

The Integer Factorization Problem

Let $N = pq$ be an RSA modulus with unknown factorization. The Integer Factorization Problem is to find p and q.

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

The RSA Problem

Given $N = pq$, e and c. Find an integer $m \in \mathbb{Z}_N^*$ such that

$$m^e \equiv c \pmod{N}.$$

RSA: The hard problems

The equations

$$N = pq, \qquad \phi(N) = (p-1)(q-1),$$
$$ed - k\phi(N) = 1, \quad c \equiv m^e \pmod{N}.$$

The Integer Factorization Problem

Let $N = pq$ be an RSA modulus with unknown factorization. The Integer Factorization Problem is to find p and q.

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

The RSA Problem

Given $N = pq$, e and c. Find an integer $m \in \mathbb{Z}_N^*$ such that

$$m^e \equiv c \pmod{N}.$$

RSA: The hard problems

The equations

$$N = pq, \qquad \phi(N) = (p-1)(q-1),$$
$$ed - k\phi(N) = 1, \quad c \equiv m^e \pmod{N}.$$

The Integer Factorization Problem

Let $N = pq$ be an RSA modulus with unknown factorization. The Integer Factorization Problem is to find p and q.

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

The RSA Problem

Given $N = pq$, e and c. Find an integer $m \in \mathbb{Z}_N^*$ such that

$$m^e \equiv c \pmod{N}.$$

RSA: The hard problems

The equations

$$N = pq, \qquad \phi(N) = (p-1)(q-1),$$
$$ed - k\phi(N) = 1, \quad c \equiv m^e \pmod{N}.$$

The Integer Factorization Problem

Let $N = pq$ be an RSA modulus with unknown factorization. The Integer Factorization Problem is to find p and q.

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

The RSA Problem

Given $N = pq$, e and c. Find an integer $m \in \mathbb{Z}_N^*$ such that

$$m^e \equiv c \pmod{N}.$$

The RSA Cryptosystem

RSA-like cryptosystems

1. The Rabin cryptosystem (1978), $N = pq$.
2. The KMOV cryptosystem, Koyama, Maurer, Okamoto, Vanstone, (1991), $N = pq$.
3. Luc cryptosystem, Smith and Lennon (1993), $N = pq$.
4. The Okamoto-Uchiyama cryptosystem, (1993), $N = p^2q$.
5. The AAβ cryptosystem, Ariffin, Asbullah, Abu, Mahad, (2012), $N = p^2q$.

Contents

Diophantine Approximations

Definition

The continued fraction expansion of a real number x is an expression of the form

$$x = [a_0, a_1, a_2, \cdots] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \mathbb{N} - \{0\}$ for $i \geq 1$.

- The numbers $a_0, a_1, a_2, \cdots$ are called the partial quotients.
- For $i \geq 0$, the fractions $\frac{r_i}{s_i} = [a_0, a_1, a_2, \cdots, a_i]$ are called the convergents.
- For $x = \frac{a}{b}$, the continued fraction algorithm computes the convergents in polynomial time.

Diophantine Approximations Based Attacks

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

Known facts

1. $N = pq$, $q < p < 2q$.
2. $\phi(N) = (p-1)(q-1) = N + 1 - p - q$.
3. $\phi(N) \in \left[N - 3\sqrt{N} < \phi(N) < N - 2\sqrt{N}\right]$.

The key equation $ed - k\phi(N) = 1 \Longrightarrow ed - kN = 1 - k(p+q-1) \Longrightarrow$

$\left|\frac{k}{d} - \frac{e}{N}\right| = \frac{|1-k(p+q-1)|}{Nd} \Longrightarrow \frac{k}{d}$ is an approximation of $\frac{e}{N}$.

Diophantine Approximations Based Attacks

Theorem (Legendre)

Suppose $\gcd(a,b) = \gcd(x,y) = 1$ *and* $\left|\frac{a}{b} - \frac{x}{y}\right| < \frac{1}{2y^2}$. *Then* $\frac{x}{y}$ *is one of the convergents of the continued fraction expansion of* $\frac{a}{b}$.

Theorem (Wiener, 1990)

If $d < \frac{1}{3}N^{1/4}$, *then* $\frac{k}{d}$ *is one of the convergents of the continued fraction expansion of* $\frac{e}{N}$.

Variants

1. Verheul and Van Tilborg, 1997: $ed - k\phi(N) = 1$.
2. Blömer and May, 2004: $ex - \phi(N)y = z$.
3. Dujella, 2004: $ed - k\phi(N) = 1$.
4. A.N., 2008: $eX - (p-u)(q-v)Y = 1$.
5. A.N., 2009: $eX - (N - (ap+bq))Y = Z$.

Diophantine Approximations Based Attacks

Theorem (Legendre)

Suppose $\gcd(a,b) = \gcd(x,y) = 1$ *and* $\left|\frac{a}{b} - \frac{x}{y}\right| < \frac{1}{2y^2}$. *Then* $\frac{x}{y}$ *is one of the convergents of the continued fraction expansion of* $\frac{a}{b}$.

Theorem (Wiener, 1990)

If $d < \frac{1}{3}N^{1/4}$, *then* $\frac{k}{d}$ *is one of the convergents of the continued fraction expansion of* $\frac{e}{N}$.

Variants

1. Verheul and Van Tilborg, 1997: $ed - k\phi(N) = 1$.
2. Blömer and May, 2004: $ex - \phi(N)y = z$.
3. Dujella, 2004: $ed - k\phi(N) = 1$.
4. A.N., 2008: $eX - (p-u)(q-v)Y = 1$.
5. A.N., 2009: $eX - (N - (ap+bq))Y = Z$.

Contents

Coppersmith's lattice based attack

Polynomial equation

Given a multivariate polynomial f and a modulus N, find a solution $(x_1, \ldots, x_n)$ of the equation

$$f(x_1, \ldots, x_n) \equiv 0 \pmod{N}.$$

Coppersmith's method

1. Lattices.
2. The LLL algorithm.
3. Jochemz-May strategy.
4. Howgrave-Graham's method.
5. Gröbner basis or resultant computation techniques.

Introduction to lattices

Definition

Let n and d be two positive integers. Let $b_1 \cdots, b_d \in \mathbb{R}^n$ be d linearly independent vectors. The lattice $\mathcal{L}$ generated by $(b_1 \cdots, b_d)$ is the set

$$\mathcal{L} = \sum_{i=1}^{d} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z} \right\}.$$

The vectors $b_1 \cdots, b_d$ are called a vector basis of $\mathcal{L}$. The lattice rank is n and the lattice dimension is d. If $n = d$ then $\mathcal{L}$ is called a full rank lattice.

Introduction to lattices

Lattice in dimension 2

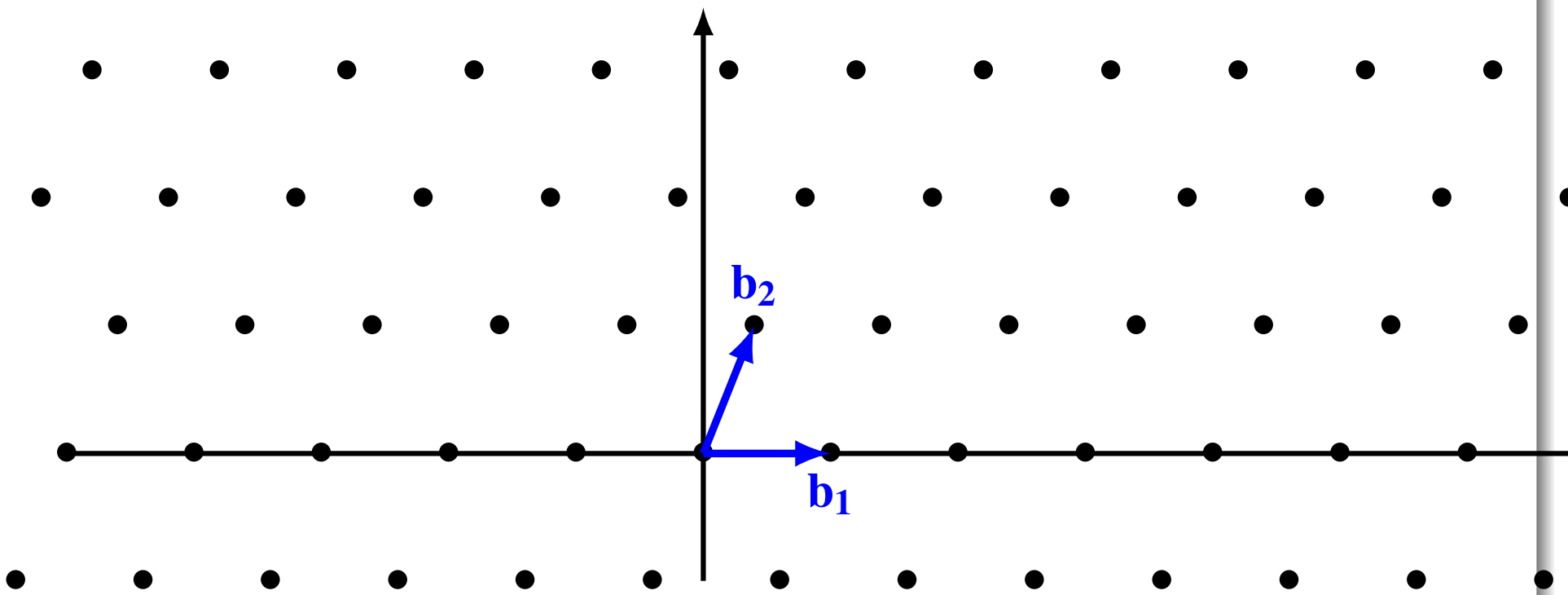


Figure: A lattice with the basis (b_1, b_2)

Introduction to lattices

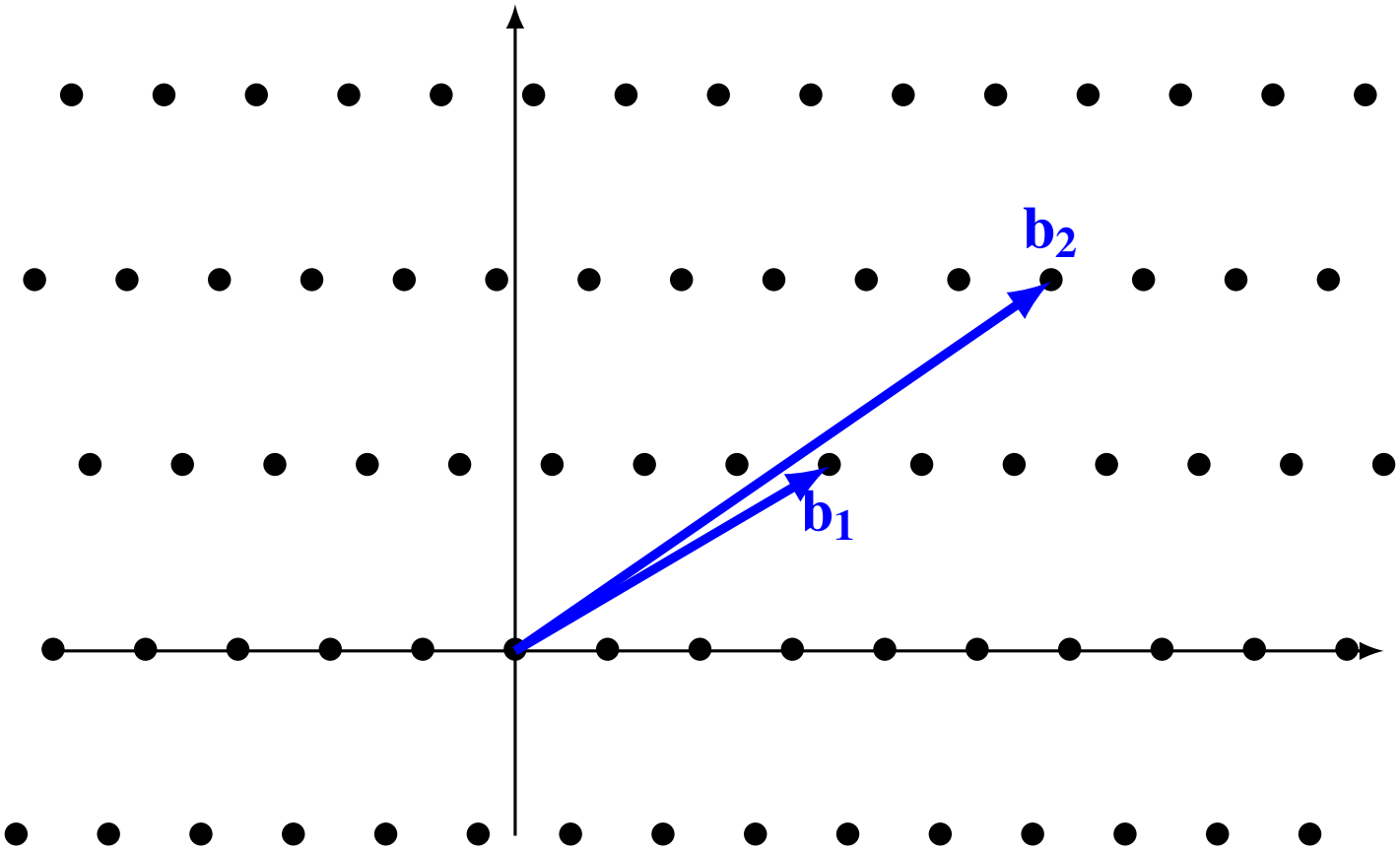


Figure: A lattice with *a bad* basis (b_1, b_2)

Introduction to lattices

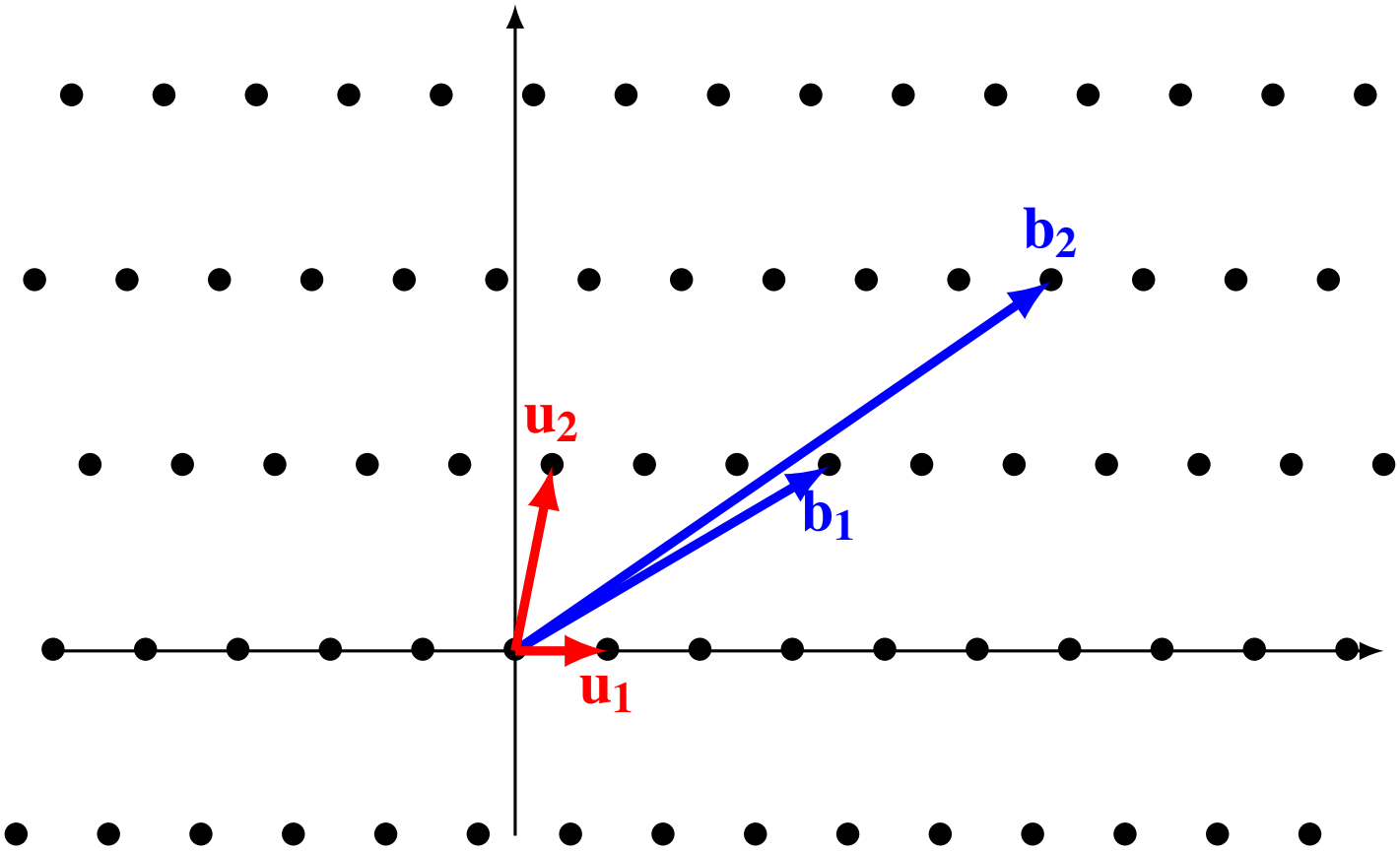


Figure: A lattice with *a good* basis (u_1, u_2)

The LLL algorithm

- Invented in 1982 by Lenstra, Lenstra and Lovász.
- Given an arbitrary basis B of a lattice $\mathcal{L}$, finds a "good" basis.
- Polynomial time algorithm.
- Various applications:
  1. Formulae for π, $\log 2$, ...

$$\pi = \sum_{i=0}^{\infty} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right).$$

  2. Implemented in Mathematica, Maple, Magma, Pari/GP, ...
  3. Solving diophantine equations.
  4. Solving SVP and CVP problems in low dimensions.
  5. Cryptanalysis of Knapsack cryptosystems.
  6. Attacks on RSA and NTRU.

The LLL algorithm

LLL-reduced basis: properties

Theorem

Let $(b_1 \cdots, b_n)$ be an LLL-reduced basis and $(b_1^*, \cdots, b_n^*)$ be the Gram-Schmidt orthogonal associated basis. We have

1. $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$ *for* $1 \leq j \leq i \leq n$.
2. $\prod_{i=1}^{n} \|b_i\| \leq 2^{\frac{n(n-1)}{4}} \det(L)$.
3. $\|b_j\| \leq 2^{\frac{i-1}{2}} \|b_i^*\|$ *for* $1 \leq j \leq i \leq n$.
4. $\|b_1\| \leq 2^{\frac{n-1}{4}} \det(L)^{\frac{1}{n}}$.
5. *For any nonzero vector* $v \in L$, $\|b_1\| \leq 2^{\frac{n-1}{2}} \|v\|$.

Coppermith's method

Theorem (Howgrave-Graham)

Let $h(x_1, \cdots, x_n) \in \mathbb{Z}[x_1, \cdots, x_n]$ *be a polynomial with at most* ω *monomials. Suppose that*

1. $h\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) \equiv 0 \pmod{R}$ *where* $|x_i^{(0)}| < X_i$ *for* $i = 1, \ldots, n$,
2. $h(x_1X_1, \cdots, x_nX_n) < \frac{R}{\sqrt{\omega}}$.

Then $h\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) = 0$ *holds over the integers.*

Coppermith's method

Polynomial equation

Given a multivariate polynomial f and a modulus N, find a solution $(x_1, \ldots, x_n)$ of the equation

$$f(x_1, \ldots, x_n) \equiv 0 \pmod{N}.$$

Principles of Coppermith's method

1. f is a polynomial with small roots.
2. Use f to build ω new polynomials sharing the roots.
3. Use the new polynomials to build a lattice $\mathcal{L}$ with a basis B.
4. Apply the LLL algorithm to reduce the basis B.
5. Solve the polynomials of the reduced basis using Howgrave-Graham's Theorem and resultant or Gröbner Basis techniques.

Applications of Coppermith's method

Theorem (Coppersmith, 1996)

Let $N = pq$ be the product of two unknown integers such that $q < p < 2q$. Given an approximation of p with additive error at most $N^{\frac{1}{4}}$, then p and p can be found in polynomial time.

Theorem (Boneh and Durfee, 1999)

Let $N = pq$ be the product of two unknown integers such that $q < p < 2q$. If d is private exponent such that $d < N^{0.292}$, then p and p can be found in polynomial time.

Many attacks on RSA are based on Coppersmith's method.

Contents

Side Channel Attacks

Various side channel attacks on RSA based on

1. Power consumption.
2. Time.
3. Magnetic emanation.
4. Acoustic vibration.
5. Faults.
6. Performing the exponentiation with the private exponent d.
7. Performing the Chinese Remainder Theorem (CRT) with the prime factors p and q.

In general, it is believed that side channel attacks cannot be used when many concurrent processes are running on the system.

Side Channel Attacks

Modular exponentiation

1. **Input**: c, d, N.
2. **Output**: $c^d \pmod N$.
3. $z = d, u = 1, v = c$.
4. While $z > 0$ do
   1. if $z \equiv 1 \pmod 2$ then $u = uv \pmod N$.
   2. $v = v^2 \pmod N$.
   3. $z = \frac{z}{2}$.
5. return u.

The time or power consumption is higher when performing the "*if*".

Side Channel Attacks

Chinese Remainder Theorem

1. **Input**: c, d_p, d_q.
2. **Output**: M.
3. Compute $M_p \equiv C^{d_p} \pmod{p}$.
4. Compute $M_q \equiv C^{d_q} \pmod{q}$.
5. Use the Chinese Remainder Theorem (CRT) to find M satisfying $M \equiv M_p \pmod{p}$ and $M \equiv M_q \pmod{q}$.
6. return M.

Contents

Recent Attacks on RSA

The attacks

1. Hininger et al, 2012.
2. Bernstein et al. attack, 2013.
3. Shamir et al. attack, 2013.
4. Ariffin et al., 2014.

Recent Attacks on RSA

Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices

Nadia Heninger[†*] Zakir Durumeric[‡*] Eric Wustrow[‡] J. Alex Halderman[‡]

Heninger et al. attack, 2012

- Efficiently factor thousands of RSA moduli in use on the Internet.
  1. 0.5 percent of 13 million TLS (Transport Layer Security) certificates share keys.
  2. 0.5 percent of 10 million SSH (Secure Shell) certificates share keys.

Recent Attacks on RSA

Factoring RSA keys from certified smart cards: Coppersmith in the wild

Daniel J. Bernstein[1,2], Yun-An Chang[3], Chen-Mou Cheng[3], Li-Ping Chou[4], Nadia Heninger[5], Tanja Lange[2], and Nicko van Someren[6]

Bernstein et al. attack, 2013

- Efficiently factor 184 distinct RSA keys out of more than two million 1024-bit RSA keys downloaded from Taiwan's national "Citizen Digital Certificate" database.
  1. 103 keys share primes.
  2. 81 keys were found by applying Coppersmith's method.
- The same prime factor was used 46 times.

Recent Attacks on RSA

Genkin, Shamir and Tromer's acoustic attack, 2013

- The authors studied the acoustic emanations emitted by computers during operations.
- The attack can extract full 4096-bit RSA decryption keys from laptop computers within an hour.
  1. Applicable to GnuPG's current implementation of RSA.
  2. Up to a distance of 30 cm using a mobile phone.
  3. Up to 4 meters using a more sensitive parabolic microphone.

Attack on RSA using k variant key equations

Theorem (Ariffin et al., 2014)

Let $(N_1, e_1), \ldots, (N_k, e_k)$ be k RSA public keys such that

$$\begin{cases} e_1 x - y_1 \phi(N_1) = z_1, \\ \quad \ldots \quad = \ldots \\ e_k x - y_k \phi(N_k) = z_k. \end{cases} \quad \text{with} \quad \begin{cases} N = \min_i N_i, \\ \delta = \frac{k}{2(k+1)}, \\ x < N^{\delta}, \\ y_i < N^{\delta}, \\ |z_i| < \frac{p_i - q_i}{3(p_i+q_i)} y_i N^{1/4}. \end{cases}$$

Then one can factor the k RSA moduli $N_1, \cdots, N_k$ in polynomial time.

Roadmap for the proof

1. Find k simultaneous diophantine approximations.
2. Use the LLL algorithm to solve the problem.
3. Use Coppersmith's technique to find the prime factors.

Attack on RSA using k variant key equations

Theorem (Ariffin et al., 2014)

Let $(N_1, e_1), \ldots, (N_k, e_k)$ be k RSA public keys such that

$$\begin{cases} e_1 x - y_1 \phi(N_1) = z_1, \\ \quad \ldots \quad = \ldots \\ e_k x - y_k \phi(N_k) = z_k. \end{cases} \quad \text{with} \quad \begin{cases} N = \min_i N_i, \\ \delta = \frac{k}{2(k+1)}, \\ x < N^{\delta}, \\ y_i < N^{\delta}, \\ |z_i| < \frac{p_i - q_i}{3(p_i+q_i)} y_i N^{1/4}. \end{cases}$$

Then one can factor the k RSA moduli $N_1, \cdots, N_k$ in polynomial time.

Roadmap for the proof

1. Find k simultaneous diophantine approximations.
2. Use the LLL algorithm to solve the problem.
3. Use Coppersmith's technique to find the prime factors.

Contents

Conclusion

1. Hundreds of attacks on RSA.
2. Most of the attacks use information of misuse of the RSA system.
3. All the attacks can be avoided.
4. The RSA system remains secure and can be trusted.
5. Just mind your Ps and Qs in the RSA modulus $N = pq$.

Terima kasih

Thank you

Merci

Danke

شكرا