Side Channel Attack to Actual Cryptanalysis: Breaking CRT-RSA with Low Weight Decryption Exponents

Santanu Sarkar and Subhamoy Maitra

Leuven, Belgium 12 September, 2012

Outline of the Talk

RSA Cryptosystem

CRT-RSA

CRT-RSA having Low Hamming Weight Decryption Exponents

The RSA Public Key Cryptosystem

- Invented by Rivest, Shamir and Adleman in 1977.
- Most popular public key cryptosystem.
- Used in Electronic commerce protocols.

RSA in a Nutshell

Key Generation Algorithm

- Choose primes p, q (generally same bit size, $q < p < 2q$)
- Construct modulus $N = pq$, and $\phi(N) = (p-1)(q-1)$
- Set e, d such that $d = e^{-1} \bmod \phi(N)$
- Public key: (N, e) and Private key: d

Encryption Algorithm: $C = M^e \bmod N$

Decryption Algorithm: $M = C^d \bmod N$

RSA and Factorization

"The primes p, q guard the secret of RSA."

- Factoring $N = pq$ implies 'attack' on RSA. [the reverse is not proved yet]
- However, as of today, factoring N is *infeasible* for $\log_2(N) > 768$
- And practical RSA uses $\log_2(N) = 1024, 2048$ (recommended)

Simple factoring of $N = pq$ does not seem to be an efficient solution!

Square and Multiply

Input: x, y, N
Output: $x^y \bmod N$

1 $z = y, u = 1, v = x$;
2 **while** $z > 0$ **do**
3 **if** $z \equiv 1 \bmod 2$ **then**
4 $u = uv \bmod N$;
  end
5 $v = v^2 \bmod N$; $z = \lfloor \frac{z}{2} \rfloor$;
end
6 return u.

Algorithm 1: The fast square and multiply algorithm for modular exponentiation.

- $\ell_y = \lceil \log_2 y \rceil$ many squares
- $w_y = wt(bin(y))$ many multiplications

Square and Multiply algorithm

Cost of calculating $x^y \bmod N$

- Squares: ℓ_y(bit length of y)
- Multiplications: $w_y \approx \frac{\ell_y}{2}$ (weight of y)
- Total Modular Multiplications: $\ell_y + w_y \approx \frac{3}{2}\ell_y$
- Total Bit Operations: $\frac{3}{2}\ell_y \ell_N^2$

The CRT-RSA Cryptosystem

- Improves the decryption efficiency of RSA, 4 folds!
- Invented by Quisquater and Couvreur in 1982.
- The most used variant of RSA in practice.
- PKCS #1 standard: store the RSA secret parameters as a tuple $(p, q, d, d_p, d_q, q^{-1} \bmod p)$.

Chinese Remainder Theorem(CRT)

Theorem

Let r, s be integers such that $gcd(r, s) = 1$. Given integers a, b, there exists unique $x < rs$ such that

1. $x \equiv a \bmod r$
2. $x \equiv b \bmod s$

CRT-RSA: Faster approach for decryption

- Two decryption exponents (d_p, d_q) where

$$d_p \equiv d \bmod (p-1) \text{ and } d_q \equiv d \bmod (q-1).$$

- To decrypt the ciphertext C, one needs

$$C_p \equiv C^{d_p} \bmod p \text{ and } C_q \equiv C^{d_q} \bmod q.$$

Calculating x^y:

- $\ell_y = \lceil \log_2 y \rceil$ many squares
- $w_y = wt(bin(y))$ many multiplications

Efficiency of CRT-RSA Decryption

- For $e = 2^{16} + 1$, we have $\ell_{d_p} \approx \ell_{d_q} \approx \frac{\ell_N}{2}$
- $C^{d_p} \bmod p$ requires $\frac{3}{2}\ell_{d_p}\ell_p^2 \approx \frac{3}{16}\ell_N^3$ many bit operation
- $C^{d_q} \bmod q$ requires $\frac{3}{2}\ell_{d_q}\ell_q^2 \approx \frac{3}{16}\ell_N^3$ many bit operation
- Total bit operations for decryption is $\frac{3}{8}\ell_N^3$

CRT-RSA: Faster through low Hamming weight

- Lim and Lee (SAC 1996) and later Galbraith, Heneghan and McKee (ACISP 2005): d_p, d_q with low Hamming weight.
- Maitra and Sarkar (CT-RSA-2010): large low weight factors in d_p, d_q.
- The security analysis of all these schemes argue that the exhaustive search for the low Hamming weight factors in the decryption exponents is the most efficient approach to attack such a scheme.

Galbraith, Heneghan and McKee (ACISP 2005)

Input: ℓ_e, ℓ_N, ℓ_k
Output: p, d_p

1 Choose an ℓ_e bit odd integer e;
2 Choose random ℓ_k bit integer k_p coprime to e;
3 Find odd integer d_p such that $d_p \equiv e^{-1} \bmod k_p$;
4 $p = 1 + \frac{ed_p - 1}{k_p}$;

$(\ell_e, \ell_N, \ell_d, \ell_k) = (176, 1024, 338, 2)$ WITH $w_{d_p} = w_{d_q} = 38$

Comparison in decryption: $\frac{2 \times \frac{3}{2} \times 338 \times 512^2}{2 \times (338+38) \times 512^2} \Rightarrow 26\%$ Faster

Security of the Algorithm

- Brute force search
- Lattice attack by May (Crypto 2002)
- Lattice attack by Bleichenbacher and May (PKC2006)
- Lattice attack by Jochemsz and May (Crypto 2007)

Security of the Algorithm

- Brute force search
- Lattice attack by May (Crypto 2002)
- Lattice attack by Bleichenbacher and May (PKC2006)
- Lattice attack by Jochemsz and May (Crypto 2007)

BUT ..

The Tool for Cryptanalysis

- Heninger and Shacham: Reconstructing RSA private keys from random key bits. Crypto 2009. Some bits are not available.
- Henecka, May and Meurer: Correcting Errors in RSA Private Keys (Crypto 2010).
- w_{d_p}, w_{d_q} are taken significantly smaller than the random case.
- Take the all zero bit string as error-incorporated (noisy) presentation of d_p, d_q.
- If the error rate is significantly small, one can apply the error correcting algorithm of Henecka et al to recover the secret key.
- Time complexity of the error-correction heuristic: τ.
- The strategy attacks the schemes of SAC 1996 and ACISP 2005 in $\tau O(e)$ time. For our scheme in CT-RSA 2010, it is $\tau O(e^3)$.

Attack Algorithm

Input: N, e, k_p, k_q and a, C
Output: Set A, containing possible guesses for p.

1 Initialize $b = 0, A = \emptyset, A_{-1} = \emptyset$;

2 **while** $b < \frac{\ell_N}{2}$ **do**

3 $\quad A = \{0,1\}^a || A_{-1}$;

4 $\quad$ For each possible options $p' \in A$, calculate $q' = (p')^{-1} N \bmod 2^{b+a}$;

5 $\quad$ For each p', q', calculate
$\quad d'_p = (1 + k_p(p' - 1))\, e^{-1} \bmod 2^{b+a}, d'_q = (1 + k_q(q' - 1))\, e^{-1} \bmod 2^{b+a}$;

6 $\quad$ If the number of 0's taking together the binary patterns of d'_p, d'_q in the positions b to $b + a - 1$ from the least significant side is less than C, then delete p' from A;

7 $\quad$ If $b \neq 0$ and $A = \emptyset$, then terminate the algorithm and report failure;

8 $\quad A_{-1} = A$; $b = b + a$;

end

9 Report A;

The Heuristic: Henecka et al

Theorem

Let $a = \lceil \frac{\ln \ell_N}{4\epsilon^2} \rceil, \gamma_0 = \sqrt{(1+\frac{1}{a})\frac{\ln 2}{4}}$ *and* $C = a + 2a\gamma_0$. *We also consider that the parameters* k_p, k_q *of CRT-RSA are known. Then one can obtain* p *in time* $O(l_N^{2+\frac{\ln 2}{2\epsilon^2}})$ *with success probability greater than* $1 - \frac{2\epsilon^2}{\ln \ell_N} - \frac{1}{\ell_N}$ *if* $\delta \leq \frac{1}{2} - \gamma_0 - \epsilon$.

- To maximize δ, ϵ should converge to zero and in such a case a tends to infinity.
- Then the value of γ_0 converges to 0.416.
- Thus, asymptotically Algorithm 3 works when δ is less than $0.5 - 0.416 = 0.084$.
- Since in this case a becomes very large, the algorithm will not be efficient and may not be implemented in practice.
- This is the reason, experimental results could not reach the theoretical bounds as studied in the work of Henecka et al.

CRT-RSA Cryptanalysis

- Following the idea of Henecka et al, one can cryptanalyze CRT-RSA having $w_{d_p}, w_{d_q} \leq 0.04\ell_N$ in $O(e \cdot \mathrm{poly}(\ell_\mathrm{N}))$ time.
- For each possible option of k_p, k_q (this requires $O(e)$ time), one needs to apply the Algorithm to obtain p.
- For small e the attack remains efficient.

Improving the Heuristic

- While applying the heuristic of Henecka et al, we noted a few modifications that can improve the performance significantly.
- Different values of the threshold
- Multiple constraints on each round

Input: $N, e, k, k_p, k_q, \tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q}, a, B$ and threshold parameters
Output: Set A, containing possible guesses for p.

1 Initialize $b = 0, A = \emptyset, A_{-1} = \emptyset$;

2 **while** $b < \frac{\ell_N}{2}$ **do**

3 $A = \{0,1\}^a \| A_{-1}$;

4 For each possible options $p' \in A$, calculate $q' = (p')^{-1} N \bmod 2^{b+a}$;

5 Calculate $d' = (1 + k\,(N + 1 - p' - q'))\, e^{-1}) \bmod 2^{b+a}$,
$d'_p = (1 + k_p(p' - 1))\, e^{-1} \bmod 2^{b+a}$, $d'_q = (1 + k_q(q' - 1))\, e^{-1} \bmod 2^{b+a}$;

6 Calculate μ_i's for $i = 1$ to 31 comparing least significant $b + a$ bits of the noisy strings and the corresponding possible partial solution strings of length $b + a$, i.e., through the positions 0 to $b + a - 1$;

7 If $\mu_i < C_i^{a+b}$ for any $i \in [1, \ldots, 31]$, delete the solution from A;

8 If $|A| > B$, reduce C_{31}^{a+b} by 1 and go to Step 7;

9 If $b \neq 0$ and $A = \emptyset$, then terminate the algorithm and report failure;

10 $A_{-1} = A$; $b = b + a$;

end

11 Report A;

Algorithm 2: Improved Error Correction algorithm.

Improving the Heuristic (Experimental Results)

	Upper bound of δ [H]		Success probability (expt.)		δ
	th.	expt.	[H]	our	our expt.
(p, q)	0.084	0.08	0.22	0.61	0.12
(p, q, d)	0.160	0.14	0.15	0.52	0.17
(p, q, d, d_p, d_q)	0.237	0.20	0.21	0.50	0.25

- We run the strategy till we obtain all the bits of p.
- It is known that if one obtains the least significant half of p, then it is possible to obtain the factorization of N efficiently

Experimental results: parameters d_p, d_q

δ	0.08	0.09	0.10	0.11	0.12	0.13
Suc. prob.	0.59	0.27	0.14	0.04	-	-
Time (sec.)	307.00	294.81	272.72	265.66	-	-
Suc. prob.	0.68	0.49	0.25	0.18	0.08	0.02
Time (sec.)	87.41	84.47	80.18	74.57	79.33	76.04

Lim et al (SAC 1996)

- $\ell_N = 768, \ell_{d_p} = 384, w_{d_p} = 30, e = 257; \Rightarrow \delta \approx \frac{30}{384} = 0.078$
- $\ell_N = 768, \ell_{d_p} = 377, w_{d_p} = 45, e = 257; \Rightarrow \delta = \frac{w_{dp}}{\ell_{dp}} \approx 0.12$

Galbraith et al (ACISP 2005)

$(\ell_e, \ell_{d_p}, \ell_{k_p}) = (176, 338, 2), w_{d_p} = 38 \Rightarrow \delta \approx \frac{38}{338} \approx 0.11$

Maitra et al (CT-RSA 2010) $\delta \approx 0.08$

Conclusion

- Application of the recently proposed error correction strategy of secret keys for RSA by Henecka et al to actual cryptanalysis. We studied two kinds of schemes.
  - CRT-RSA decryption keys are of low weight as (SAC 1996, ACISP 2005). We demonstrate complete break in a few minutes for 1024 bit RSA moduli.
  - The decryption exponents are not of low weight, but they contain large low weight factors (CT-RSA 2010). Actual break is not possible, but clear cryptanalytic result.

- We had a detailed look at the actual error correction algorithm of Henecka et al.
  - We provide significant improvements as evident from experimental results.
  - We could demonstrate that the theoretical bound given by Henecka et al can also be crossed using our heuristic.


Thank You!