Implementation of RSA Attack Using 2-Dimensional Lattices by Constructing Hypotheses of Keys With Low Hamming Weight

Miroslav Dimitrov, Tsonka Baicheva
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
Bulgaria
miroslavdimitrov@fmi.uni-sofia.bg, tsonka@math.bas.bg
Georgi Ivanov
State Agency "National Security"
Bulgaria

***Abstract*—We describe an attack on the RSA cryptosystem, under the assumption that the Hamming weight of a sufficiently long sequence of bits of the MSBs and/or LSBs of the private exponent d is low. The attack uses 2-Dimensional lattices and has been implemented in SageMath.**

1. Introduction

RSA is one of the first public-key cryptosystems and it is currently widely used by providing means of secure data transmission. It can be efficiently implemented on constrained devices such as smartcards. By using powerful attacks such as side-channel attacks, power analysis, timing attacks and others, an attacker can partially expose bits of the private exponent d or construct a hypothesis about the Hamming weight of consequent bits of d.

The concept of partial key exposure attacks on RSA was introduced in 1997 by Boneh, Durfee and Frankel in [1]. They showed that for low exponent RSA, a quarter of the list significant bits of d are sufficient for efficiently reconstructing d and obtained similar results for larger values of e as long as $e < \sqrt{N}$. These results demonstrate the danger of leaking part of the bits of d. Later Blömer and May [2] extended their result with attacks for $e \in \left[N^{0.5}, N^{0.725}\right]$. Ernst, Jochemsz, May, and De Weger [3] have constructed flexible attacks including the cases where the private exponent d or the public exponent e is chosen to be small and both their attacks work up to full size exponents.

In [2], [3], [4], [5] lattice methods are used to perform a partial key exposure attack. The attacks using lattice methods are asymptotic, meaning that if one comes close to the maximal value for the unknown part of d for which an attack should work, the lattices involved are very large. This implies that the lattice reduction phase, for which the LLL-algorithm [6] is used, may take a prohibitively long time. Therefore, it may be useful to look at very small lattices instead of very large.

In [7] the authors explore the sizes of d for which one can mount an attack in a few seconds with a very simple method using a 2-dimensional lattice. More precisely, the attack can be applied when the private exponent d is chosen to be small, which can occur in practice. Although the attack does not achieve the theoretic bounds of known partial key exposure attacks using Coppersmith's method, it is much faster in its application range.

In our work, we provide an automatic way of searching through possible choices of the private exponent d, under the condition that a sufficient length of consequent bits of the MSBs and/or LSBs of d has low Hamming weight. Moreover, even knowing only the public key, the attack can be used to blindly test different lengths d of the private exponent or to systematically search possible low Hamming weights of the first, the last or some combination of the first and the last $(2\beta-1/2)n$ bits of d for $0 < \beta < \frac{1}{2}$. The search is blind because the hypothesis intervals for some expected values of d are not given as well as the possible Hamming weights for the corresponding part of d are not determined.

For speeding up our search, we have used compact 2-dimensional Lattices only, as well as non-intersecting search spaces, which provide an additional possibility for parallel computation. The attack is implemented in SageMath.

2. Preliminaries

2.1. RSA algorithm

According to [8], we first compute n as the product of two primes p and q, i.e. $n = p * q$. These primes are sufficiently large "random" primes, and although we make n public, no efficient, non-quantum integer factorization algorithm is known. We then choose an integer d, which is an efficiently large random integer relatively prime to $(p-1)*(q-1)$. The integer e is calculated as follows:

$$e * d \equiv 1 \ (\mathsf{mod}\ (p-1)*(q-1))$$

Encrypting a message M is done by raising it to the e-th power modulo n. To decrypt a ciphertext C, we raise it to power d, again modulo n, i.e.:

$$C \equiv E(M) \equiv M^e \ (\mathsf{mod}\ n)$$

$$D(C) \equiv C^d \equiv M^{e*d} \equiv M \pmod{n}$$

The pair (n, e) is the public key and the pair (n, d) is the corresponding private key. Furthermore, we define e and d as public and private exponents.

2.2. Lattice

Let $v_1, ..., v_n \in Z^m$, $m \geq n$ be linearly independent vectors. A lattice L spanned by $\{v_1, ..., v_n\}$ is the set of all integer linear combinations of $v_1, ..., v_n$, such that

$$L = \left\{ v \in Z^m \mid v = \sum_{i=1}^{n} a_i v_i \text{ with } a_i \in Z \right\}.$$

The set of vectors $B = \{v_1, ..., v_n\}$ is called a basis of L. We also say that L is spanned by the vectors of the basis B. We define the dimension of L as $dim(L) := n$. The example in figure 1 presents a 2-dimensional lattice with $B = \{(0, 1), (1, 0)\}$.

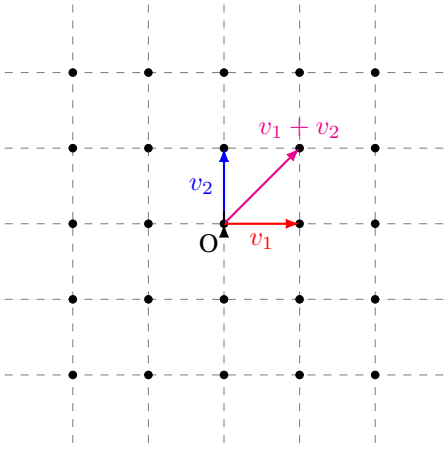

Figure 1. A 2-dimensional lattice example.

Theorem 1. (Lenstra, Lenstra, Lovász. [9])
Let $L \in Z^n$ be a lattice spanned by $B = \{v_1, ..., v_n\}$. The L^3-algorithm outputs a reduced lattice basis $\{v_1, ..., v_n\}$ with

$$\|v_i\| \leq 2^{\frac{n(n-1)}{4(n-i+1)}} \, det(L)^{\frac{1}{n-i+1}} \quad for \; i = 1, ..., n$$

in time polynomial in n and in the bit-size of the entries of the basis matrix B.

2.3. A Partial Key Exposure Attack

In [7] it is shown how to perform a partial key exposure attack on RSA using a 2-dimensional lattice. The attack applies when the private exponent d is chosen to be small, which occurs in practice. The result can be summarized as follows.

Assumption 1. The reduced basis vectors of a 2-dimensional lattice have a norm $\approx det(L)^{\frac{1}{2}}$.

Under a reasonable heuristic assumption that we specify in Assumption 1, the following holds.

Theorem 2. Let $N = pq$ be an n-bit RSA-modulus, and let p,q be primes of bitsize $\frac{n}{2}$. Let $0 < \beta < \frac{1}{2}$, and let e, d satisfy $ed \equiv 1 \; mod \; \phi(N)$ with $bitsize(e) = n$ and $bitsize(d) = \beta n$. Given an amount (total) of $(2\beta - \frac{1}{2})n$ MSBs and/or LSBs of d, N can be factored very efficiently, using a 2-dimensional lattice.

3. Attack by constructing hypotheses of the MSBs of d

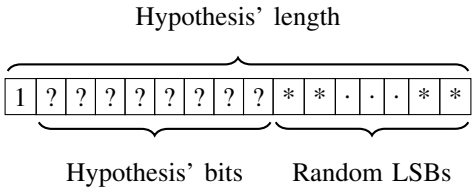

Figure 2. An overview of the search space

The algorithm takes as inputs the public key pair (n, e), the hypothesis about the bit-length of the private key (see Fig. 2), as well as the expected Hamming weight of the first s MSBs of d. Both, the bit-length of the private key and the expected low Hamming weight of the first s MSBs of d, can be given as a number interval. The outer cycle of the algorithm iterates through all given bit-lengths of the private key d, each yielding the value of β. Having β we can deduct the required amount S of MSBs of d to successfully apply a 2-dimensional lattice attack. The inner cycle of the algorithm iterates through all possible combinations of the first s MSBs of d, such that the Hamming weight of s is equal to the current Hamming weight. If $s < S$, the algorithm can try to brute-force the remaining $S - s$ bits (which is a reasonable approach if $S - s$ is small enough). In the case when $s \geq S$, we initiate the core of the algorithm.

4. Validating the Hypothesis

Let $d = N^\beta < N^{\frac{1}{2}}$ and $e < \phi(N) < N$. Let the "unknown middle part" of d is of size N^δ with $\delta < \frac{1}{2} - \beta$. Let d_L be the known LSB part of d of size N^k, followed by an unknown middle part x of size N^δ, followed by the known MSB part d_M of size $N^{\beta-k-\delta}$. Therefore, $d = d_L + 2^{\lfloor kn \rceil} x + 2^{\lfloor kn \rceil + \lfloor \delta n \rceil} d_M$, where $\lfloor \rceil$ is the nearest integer. By substituting the partition d in the RSA key equation, we must find the solution $(x, y, z) = (x, k, p + q - 1)$ of the equation

$$e2^{\lfloor kn \rceil} x - Ny + yz + R - 1 = 0,$$

$$with \; R = ed_L + e2^{\lfloor kn \rceil + \lfloor \delta n \rceil} d_M$$

The equation above implies that $|e2^{\lfloor kn \rceil}x - Ny + R| \leq 3N^{\beta+\frac{1}{2}}$. This is an inhomogeneous diophantine approximation problem for the variables x and y. To solve it, we define a lattice L spanned by the columns of G, with

$$G = \begin{pmatrix} C & 0 \\ e2^{\lfloor kn \rceil} & N \end{pmatrix} \text{ and } v = \begin{pmatrix} 0 \\ R \end{pmatrix} \quad (1)$$

where C is a suitable integer of size $N^{\beta-\delta+\frac{1}{2}}$. The lattice point $G \begin{pmatrix} x \\ -y \end{pmatrix}$ is close to v. Our strategy to find x and y is therefore to start with a lattice vector v' close to v, and add small multiples of the reduced basis to the columns of G, and obtain a reduced matrix G_{red}, whose columns still span L. The algorithm aims to find an integer pair (z_1, z_2) for which

$$G_{red} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = G \begin{pmatrix} x \\ -y \end{pmatrix} - G_{red} \left\lfloor G_{red}^{-1} v \right\rceil \quad (2)$$

Each pair (z_1, z_2) leads to pair $(x, -y)$. If we substitute x as the unknown part of d, and y as k, we can find ϕ that satisfies $ed-1 = k\phi$. We further test if the value $\frac{ed-1}{k}$ is an integer number. More details and proof of the core of the algorithm can be found in [7].

5. Conclusions

In the implementation of the RSA attack we have used the LLL-algorithm to extract G_{red}. The same algorithm can be used to extend the attack with lattices of higher (but still efficiently small) dimensions. Also, the algorithm's outer layer can be easily modified to search for specific structures/hypothesis.

As an implementation of the proposed attack, we were able to successfully solve one of the level III challenges published in "Mystery Twister C3" - the Crypto Challenge Contest [10]. As the authors stated, these challenges require a thorough background in cryptanalysis and usually significant computational power as well. The problems in this level represent current research topics that are believed to be very difficult to solve. Thus, practical solutions may not even exist and ready-to-run tools almost certainly do not. The methodology to solve some of these challenges may already be known, but it may require a huge amount of computational power, and thus only a large group of people working together in a distributed system might obtain the solution.

The particular challenge we have solved is about reconstructing the private key which corresponds to a given public key, knowing that the bit length of d is 400 bits and the Hamming weight of the first 310 bits is 4. We were able to fully recover the private key for less than 4 hours on a single computer using unoptimized source code written in SageMath[1]. The current version of the software allows the user to apply successful attacks on RSA instances, even when the partial information is so limited, that directly applying the currently known state-of-the-art RSA attacks on the given RSA instance is not applicable. The software can be used for a posteriori cryptanalysis on RSA certificates with suspiciously large public exponents or used as a fast automatic way for probing hypotheses constructed by side-channel attacks, power-analysis, timing attacks or by some other methods.

Acknowledgments

This research was partially supported by the Bulgarian National Science Fund [Contract No. 12/8, 15.12.2017].

References

[1] D. Boneh, G. Durfee and Y. Frankel, "An Attack on RSA given a Small Fraction of the Private Key Bits", *Proceedings of ASIACRYPT* 1998, LNCS vol. 1514, pp. 25-34.

[2] J. Blömer, A. May, "New Partial Key Exposure Attacks on RSA", *Proceedings of CRYPTO 2003*, LNCS vol. 2729, pp. 27-43.

[3] M. Ernst, E. Jochemsz, A. May, and B. de Weger, "Partial Key Exposure Attacks on RSA up to Full Size Exponents", *Proceedings of EUROCRYPT 2005*, LNCS vol. 3494, pp. 371-386.

[4] D. Boneh, G. Durfee, "Cryptanalysis of RSA with Private Key d less than $N^{0.292}$", *IEEE Transactions on Information Theory*, vol. 46, pp. 1339-1349, 2000.

[5] A. May, "New RSA Vulnerabilities Using Lattice Reduction Methods", Ph. D. dissertation, University of Paderborn, Oct. 2003.

[6] A. Lenstra, H. Lenstra, Jr., and Lászó Lovász, "Factoring Polynomials with Rational Coecients", *Mathematische Annalen*, vol. 261, pp. 515-534, 1982.

[7] E. Jochemsz, B. de Weger, "A Partial Key Exposure Attack on RSA Using a 2-Dimensional Lattice", in *Information Security*, S. K. Katsikas, J. Løpez, M. Backes, S. Gritzalis, B. Preneel, Eds. Springer, Berlin, Heidelberg, 2006, LNCS vol. 4176, pp. 203-216.

[8] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, vol. 21, issue 2, pp. 120-126, Feb. 1978.

[9] A. May, "Using LLL-Reduction for Solving RSA and Factorization Problems", in *The LLL-algorithm*, P. Nguyen, B. Vallée, Eds. Springer, Berlin, Heidelberg, 2009, ISC book series, pp. 315-348.

[10] "Finding Special Private RSA Keys", in *Mystery Twister C3* - the Crypto Challenge Contest, [Online]. Available: https://www.mysterytwisterc3.org/images/challenges/mtc3-kitrub-04-speciald-en.pdf.

[11] W. Stein, "SAGE: A Computer System for Algebra and Geometry Experimentation", [Online]. Available: https://wstein.org/sage.html

1. SageMath [11] is a computer algebra system with features covering many aspects of mathematics, including algebra, combinatorics, graph theory, numerical analysis, number theory, calculus and statistics. SageMath uses a syntax resembling Python's supporting procedural, functional and object-oriented constructs.