Practical Lattice-Based Fault Attack and Countermeasure on SM2 Signature Algorithm

Weiqiong Cao[1(✉)], Jingyi Feng[1], Shaofeng Zhu[1], Hua Chen[1], Wenling Wu[1], Xucang Han[2], and Xiaoguang Zheng[2]

[1] Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
{caowq,fengjingyi,zhushaofeng,chenhua,wwl}@tca.iscas.ac.cn
[2] Beijing Key Laboratory of RFID Chip Test Technology, CEC Huada Electronic Design Co., Ltd, Beijing 102209, People's Republic of China
{hanxc,zhengxg}@hed.com.cn

Abstract. We present a practical lattice-based fault attack against SM2 signature algorithm in a smart card. This seems to be the first combination of the lattice attack presented in SAC'2013 and fault attack against SM2 in practice. We successfully utilize the laser fault attack to skip the instructions of nonces being written into RAM, so that the nonces in signatures share partial same bits from each other. Next, we build the model of lattice attack and recover the private key. The experimental results show we only need 3 faulty signatures to mount lattice attack successfully in about 32 μs. Moreover, we propose a new countermeasure for SM2 signature algorithm to resist lattice-based fault attack by destroying the condition of lattice attack rather than thwarting fault attack. It is proved the countermeasure can guarantee the ability to resist lattice attack, even if some information of the nonces is leaked.

Keywords: Fault attack · Lattice attack · Countermeasure · SM2

1 Introduction

Elliptic curve cryptosystem (ECC) has been used widely in cryptographic devices such as smart card. For the implementation of ECC in device, we must analyze not only its mathematic security but also the ability against physical attacks, such as fault attack (FA). So far, there have been many results about FA against ECC [1–3], especially against elliptic curve digital signature algorithm (ECDSA) [4–6]. Among them, lattice-based fault attack (LFA) is one of the most effective attacks. It combines both fault attack (FA) and lattice attack (LA). Firstly, some information about the nonce k in signature is revealed by FA. Next, with the leakage information of k, LA can disclose the private key d_A. LAs against (EC) DSA-like signature algorithm are mainly classified into three

S. Qing et al. (Eds.): ICICS 2015, LNCS 9543, pp. 62–70, 2016.
DOI: 10.1007/978-3-319-29814-6_6

types. The first type [7–9] is based on knowing parts of the nonce k. The second type [10] is based on the fact that there exist a few same blocks in each nonce k. The last type [11] is based on the condition that some different nonces in signatures share partial same bits from each other. Nevertheless, it seems only the first type of LA combined with FA is applied successfully on ECDSA-like signature algorithm in practice [4,6,12]. There seems to be no FA combined with the other two types of LA in practice. It is worthy to do further research about the kind of FA and the corresponding countermeasures on ECDSA-like signature, such as SM2 signature algorithm.

SM2 signature algorithm (hereafter SM2) is a signature algorithm standard based on elliptic curve published by Chinese Government [13], and has been extensively used in cryptographic device in finance. The first type of LA based on knowing parts of k against SM2 has been introduced in INSCRYPT'2014 [9].

Our Contributions. A practical LFA against SM2 is presented based on the condition of LA that there are some bits shared between different nonces, and the attack is mounted in a smart card successfully. It seems that it is the first time to combine FA and the last type of LA (sharing some bits between nonces) against SM2 in practice. We first utilize practical laser FA to make the instructions of writing nonces into RAM skipped deliberately, so that the nonces in SM2 share partial same bits. Next, based on the faulty results of the above FA, we build the model of LA proposed in SAC'2013 [11] and recover the private key d_A successfully. At last, we propose a new countermeasure for SM2 to resist LFA by destroying the condition of LA directly. We also prove its security against LFA. Even if some information of the nonce k is leaked, the countermeasure still guarantees the ability to resist LA.

The remainder of the paper is organized as follows: Sect. 2 gives a brief introduction of SM2 and the basic theory of lattice. In Sect. 3, the practical LFA against SM2 is described. In Sect. 4, the countermeasure to resist LFA is presented. Finally, the conclusion is given in Sect. 5.

2 Preliminaries

2.1 SM2 Signature Algorithm

For simplicity, we only analyze the elliptic curve $E(a, b)$ in prime finite field F_p defined by the Weierstrass equation $y^2 = x^3 + ax + b \bmod p$, where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \bmod p$. The set of points on $E(a, b)$ and the infinity point $\mathcal{O}$ constitute an additive group $E(F_p)$. The scalar multiplication (SM) $Q = kG$ is the most important operation in $E(F_p)$, where $G, Q \in E(F_p)$ and $k \in \mathbb{Z}$. The detailed introduction about SM on $E(F_p)$ can be found in [14].

In SM2, the curve parameters a, b, p and the base point $G \in E(F_p)$ with order n are all given. The private key d_A is randomly selected in interval $[1, n-1]$ and the corresponding public key P_A satisfies $P_A = d_A G$.

Signature: sign message M with private key d_A.

1. Compute $e = SHA(Z_A||M)$, where $SHA(.)$ is the hash algorithm SM3 and Z_A is the public user information;
2. Select $k \in [1, n-1]$ randomly;
3. Compute $Q(x_1, y_1) = kG$;
4. Compute $r = e + x_1 \bmod n$. If $r = 0$ or $r + k = n$ then goto step 2;
5. Compute $s = (1 + d_A)^{-1}(k - rd_A) \bmod n$. If $s = 0$ then goto step 2;
6. Return results (r, s).

Verification: verify (M', r', s') with public key P_A.

1. If r' or $s' \notin [1, n-1]$ then return false;
2. Compute $e' = SHA(Z_A||M')$;
3. Compute $t = r' + s' \bmod n$. If $t = 0$ then return false;
4. Compute $(x'_1, y'_1) = s'G + tP_A$;
5. Compute $R = e' + x'_1 \bmod n$. If $R = r'$ then return true, else return false.

2.2 Lattice Attack Basis

Suppose that there exist the vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_N \in \mathbb{Z}^n$ which are all linearly independent from each other. Let $L = \{\sum_{i=1}^{N} x_i \boldsymbol{b}_i | x_1, \ldots, x_N \in \mathbb{Z}\}$, then L is the called **integer lattice** generated by the $\boldsymbol{b}_i$'s, where the vector set $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N\}$ is a basis of L. Let matrix $A = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)^T$, then for any vector $w \in L$, there exists $\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathbb{Z}^n$ satisfying $w = \boldsymbol{x}A$.

The closest vector problem (CVP): given a basis B of L and a vector $\boldsymbol{u} \in \mathbb{Z}^n$, find a lattice vector $\boldsymbol{v} \in L$ satisfying $\|\boldsymbol{v} - \boldsymbol{u}\| = \lambda(L, \boldsymbol{u})$, where $\lambda(L, \boldsymbol{u})$ is the closest distance between L and $\boldsymbol{u}$. CVP can be solved in polynomial time by the combination of LLL algorithm [15] and Babai's Nearest Plane algorithm [16]. Moreover, as presented in [7,16], it has been proved, as long as the unknown lattice vector $\boldsymbol{v} \in L$ and any nonzero vector $\boldsymbol{u} \in \mathbb{Z}^n$ satisfy$||\boldsymbol{v} - \boldsymbol{u}||^2 \leq c_1 c_2 \Delta(A)^{2/N}$, then $\boldsymbol{v}$ can be determined uniquely in polynomial time as a CVP. Here $c_1 \approx 1$, $1 < c_2 \leq N$, and $\Delta(A)$ is the determinant of matrix A.

3 Lattice-Based Fault Attack on SM2

In this section, we will introduce the procedure of the lattice-based fault attack against SM2. First, during implementing signatures repeatedly, we will mount laser fault attack (FA) on the smart card to obtain some shared bits between different nonces. Next, based on the faulty signatures derived from FA, we can build the model of the last type of lattice attack (LA) and recover the private key d_A by some known LA tools.

3.1 Experimental Condition

In the experiment, SM2 is implemented in a smart card and there are no countermeasures. The CPU frequency is 14 MHz and the bus width is 32 bits. The implementation of SM2 is based on hardware and software with a key length of 256 bits. As shown in Fig. 1, CPU implements the instructions of SM2 algorithm in EEPROM, with the help of coprocessor which supports the operations of modular addition, reduction and multiplication of big number. The random number generator is responsible for generating random numbers and sending them to RAM. In addition, as shown in Fig. 2, we use the laser attack platform of Riscure Company in experiments. Finally, the LA is performed in a computer with Inter Core i7-3770 at 3.4 GHz.

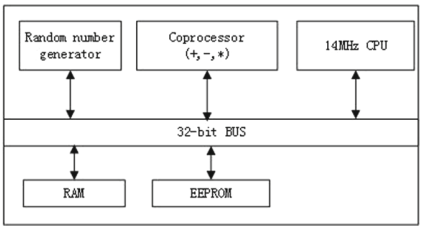

Fig. 1. The construction of smart card chip

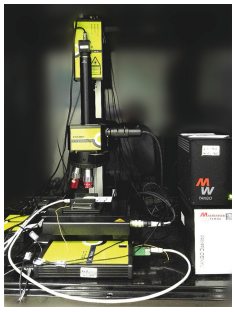

Fig. 2. The laser attack platform for FA

3.2 Fault Attack Against SM2

Actually, during the execution of SM2 in the chip, at the moment when the nonces are written into RAM through BUS, there will be obvious peak value appearing in the power consumption curve. As shown in Fig. 3, the part of power consumption curve in red box is easy to be distinguished, which indicates two operations, generating the random numbers and storing them into RAM.

For a 256-bit nonce k with big endian storage pattern, only 32 bits of k are generated every time, so it needs to generate 8 times. After each 32-bit random number is generated, it is transferred into RAM through 32-bit BUS, so we can mount laser FA at the above appropriate time. We use laser attack platform to induce forcibly some faults at the time that the random numbers will be written into RAM, so that the written instructions are skipped. As a result, the new generated random numbers are not written into RAM successfully, and the corresponding block of k remains unchanged as the last one stored in the RAM which is the block of nonce in the last signature or the initial value in RAM.

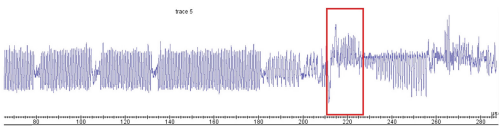

Fig. 3. The power consumption curve when the random numbers are generated and transferred (Color figure online)

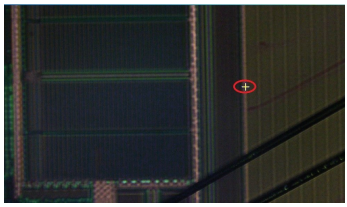

Fig. 4. The right position in EEPROM for laser fault attack (Color figure online)

This implies that there exist some bits shared between different nonces although they are still unknown.

In addition, in order to determine the right position for attack in the chip, we use the laser attacker platform to scan all the areas of the chip. Since we can not judge whether the nonces own the faults we want, we first import a known private key into the chip. Thereby, we can derive the values of all the nonces from the faulty signature results. As shown in Fig. 4, according to the derived values of nonces, the right position is determined in the red section at the edge of EEPROM. Given the proper parameters such as glitch length, laser intensity, laser duration time and so on, the success rate for obtaining some shared bits between nonces at the position is approximately 100 %. In view of the big endian storage pattern of k, we find out that the shared bits are the most significant bits (MSBs) of nonces and the number l of shared bits is a multiple of 32. After that, we import an unknown private key for real experiments. Based on the determined injection position and time, the fault attack can be mounted against many signatures uninterruptedly. Finally, we obtain 50 continuous faulty signature results $(r_i, s_i)(i = 0, \ldots, N)$.

3.3 Model of Lattice Attack Against SM2

As mentioned before, we have $N+1(N = 49)$ faulty signature results $(r_i, s_i)(i = 0, \ldots, N)$. Knowing that at least l MSBs of all the nonces are same, we can build the model of LA as presented in [11]. Let k_i and a represent respectively the nonce in the i-th signature and the shared l MSBs of all the nonces, then $k_i = a2^{m-l} + b_i (i = 0, \ldots, N)$ and $0 < a < 2^l$. Here $m(m = 256)$ is the key length of SM2 and b_i is the rest of k_i satisfying $0 < b_i < 2^{m-l}$. For $i = 0, \ldots, N$, substitute $k_i = a2^{m-l} + b_i$ into step 5 in SM2 signature, and obtain $N + 1$ equations. Then subtract the 0-th equation from the other equations respectively, and obtain the following equations.

$$(s_i + r_i - s_0 - r_0)d_A - (s_0 - s_i) = b_i - b_0 \bmod n (i = 1, \ldots, N) \tag{1}$$

Since $0 < b_i, b_0 < 2^{m-l}$, apparently $0 < |b_i - b_0| < 2^{m-l}$. Let $\Delta b_i = |b_i - b_0|$, $\Delta t_i = (s_i + r_i - s_0 - r_0) \bmod n$ and $\Delta u_i = (s_0 - s_i) \bmod n$, then the above

equations are written as

$$0 < |\Delta t_i d_A - \Delta u_i + n h_i| = \Delta b_i < 2^{m-l} (i = 1, \ldots, N). \quad (2)$$

Where $h_i \in \mathbb{Z}$ is the smallest integer which makes the above equations true. Let matrix $A = \begin{pmatrix} 1 & 2^l \Delta t_1 & \cdots & 2^l \Delta t_N \\ 0 & 2^l n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2^l n \end{pmatrix}$, then all the row vectors $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_N$ of A generate a lattice L, where $A = (\boldsymbol{b}_0, \ldots, \boldsymbol{b}_N)^T$. For $\boldsymbol{x} = (d_A, h_1, \ldots, h_N) \in \mathbb{Z}^{N+1}$, $\boldsymbol{v} = \boldsymbol{x}A = (d_A, d_A 2^l \Delta t_1 + h_1 2^l n, \ldots, d_A 2^l \Delta t_N + h_N 2^l n)$ is a nonzero lattice vector in L. Let vector $\boldsymbol{u} = (0, 2^l \Delta u_1, \ldots, 2^l \Delta u_N) \in \mathbb{Z}^{N+1}$, then the above inequations can be rewritten as $||\boldsymbol{v} - \boldsymbol{u}|| \leq 2^m \sqrt{N+1}$. As mentioned in Sect. 2, if $2^m \sqrt{N+1} \leq \sqrt{c_1 c_2} (2^{lN} n^N)^{1/(N+1)} (2^{m-1} < n < 2^m)$, i.e., $N > m/(l-1)$, then vector $\boldsymbol{v}$ can be determined uniquely by solving CVP [7], where $c_1 = N+1$ and $c_2 = 1$. Naturally, the private key d_A can be recovered from $\boldsymbol{v}$.

3.4 Attack Results

In the attack experiments, we set the length $l = 32i (i = 1, \ldots, 7)$ of the shared MSBs and the number $N = 50 - t (t = 0, \ldots, 49 - 256/l)$ of signatures by increasing the values of i and t in turn. After each setting, based on fplll-4.0 Lattice Reduction Library [17], we perform $51 - N$ attacks, where the N signatures in the i-th attack are selected from i-th signature to $(i + N - 1)$-th signature. If only one of the attacks can obtain the key d_A that meets $P_A = d_A G$, we think the lattice attack is successful.

The experimental results show that the lattice attack is still successful when $l = 192 (i = 6)$ and $N = 3 (t = 47)$, and the average time for each attack is about 32 μs. It implies that there are 192 MSBs shared between the nonces and we only need 3 signatures to disclose d_A successfully. Moreover, there are 20 successful cases in the 48 attacks with success rate 41 %.

4 Countermeasure to Resist Lattice-Based Fault Attack

In this section, a new countermeasure to resist LFA is proposed for SM2. It destroys directly the conditions of LA rather than purely preventing FA. Therefore, even though the FA has made the nonces known partially or sharing some bits, the LA still cannot be mounted.

4.1 SM2 with Countermeasure

Signature with Countermeasure: sign message M with keys d_A and P_A.

1. Compute $e = SHA(Z_A || M)$;
2. Select $k, w \in [1, n-1]$ randomly;

3. Compute $Q(x_1, y_1) = kG + wP_A$;
4. Compute $r = e + x_1 \bmod n$. If $r = 0$ or $r + k = n$ then goto step 2;
5. Compute $s = (1 + d_A)^{-1}(k + (w - r)d_A) \bmod n$. If $s = 0$ then goto step 2;
6. Return results (r, s).

In the SM2 with countermeasure above, there are two nonces k and w generated with same length, and the public key P_A is employed for signature. The nonce k is actually added with the mask wd_A. Moreover, the verification without any modification can be passed successfully.

4.2 Provable Security Against Lattice Attack

As mentioned above, the condition of the LA based on knowing parts of nonce k is strongest. Hence, it is sufficient to only analyze the security of our countermeasure against the strongest LA. Obviously, the result of analysis can be also applied similarly to our proposed attack.

As shown in SM2 Signature with countermeasure, it is assumed that we obtain $N(N > m/(l-1))$ signature results $(r_i, s_i)(i = 1, \ldots, N)$, and both the l MSBs a_i of nonce k_i and the l MSBs c_i of nonce w_i are known in the i-signature. Here m is the key length of SM2. Let b_i, d_i represent the remaining unknown values of k_i, w_i respectively, then $k_i = a_i 2^{m-l} + b_i$ and $w_i = c_i 2^{m-l} + d_i$. Where $a_i, c_i < 2^l$ and $b_i, d_i < 2^{m-l}$. Let $t_i = (s_i - c_i 2^{m-l} + r_i) \bmod n$ and $u_i = (a_i 2^{m-l} - s_i) \bmod n$, then we have the following equations.

$$t_i d_A - u_i + h_i n = b_i + d_i d_A (i = 1, \ldots, N) \tag{3}$$

where h_i is the smallest integer which makes the above equations true.

Similarly, we can construct a lattice L by matrix $A = \begin{pmatrix} \beta & t_1 & \cdots & t_N \\ 0 & n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n \end{pmatrix}$, where β is any nonzero real number. Let vector $\boldsymbol{u} = (0, u_1, u_2, \ldots, u_N) \in \mathbb{Z}^{N+1}$ and lattice vector $\boldsymbol{v} = \boldsymbol{x}A = (\beta d_A, d_A t_1 + h_1 n, \ldots, d_A t_N + h_N n)$, where $\boldsymbol{x} = (d_A, h_1, h_2, \ldots, h_N) \in \mathbb{Z}^{N+1}$, then we have the following equation

$$||\boldsymbol{v} - \boldsymbol{u}||^2 = (\beta d_A)^2 + \sum_{i=1}^{N} (b_i + d_i d_A)^2. \tag{4}$$

It is known that d_A is a random number less than $n(2^{m-1} < n < 2^m)$. The probability is $\frac{1}{2^{d+1}}$ when $\log_2 d_A = m - d$, where d is non-negative integer. Therefore, $\log_2 d_A$ is slightly smaller than or equal to $\log_2 n$. In other words, d is very small in general. In addition, $\log_2 d_i$ is much greater than d in practical FA, otherwise d_i and d_A can be directly obtained from exhaustive search attack rather than LA. Thereby, the inequation $\log_2 n - \log_2 d_A \ll \log_2 d_i$ holds, namely, $n \ll d_i d_A < n^2/2^l$. Therefore, the following inequation holds.

$$||\boldsymbol{v} - \boldsymbol{u}||^2 \gg (N+1)(n^{2N} \beta^2 d_A^2)^{1/(N+1)} > (N+1)\Delta(A)^{2/(N+1)} \tag{5}$$

As described in Sect. 2, the above inequation does not satisfy the condition of CVP, so $\boldsymbol{v}$ can not be determined. Apparently, it is also impossible to recover d_A. The conditions of LA are destroyed completely. In addition, if the known bits in the nonces are the least significant bits or contiguous blocks in the middle, the same conclusion can be proved by the similar way as above.

5 Conclusion

In this paper, we introduce a lattice-based fault attack (LFA) against SM2 in a smart card. The attack is based on the condition of lattice attack (LA) that there are some bits shared between different nonces. First, the instructions of writing nonces into RAM are skipped by practical laser fault attack (FA), so that some bits between the nonces in SM2 remain unchanged. Then we combine the results of FA with the model of LA to recover the private key d_A successfully. The experimental results show that 3 faulty signatures are needed to recover d_A with average time 32 μs and success rate 42 %. In addition, we also propose a countermeasure for SM2 to resist LFA by destroying the condition of LA from algorithm level. It is proved in theory that the countermeasure is sufficient to resist LFA. Moreover, the similar attack and countermeasure can also be applied to ECDSA.

Acknowledgments. We thank the anonymous referees for their careful reading and insightful comments. This work is supported by the National Science and Technology Major Project (No. 2014ZX01032401-001) and the National Basic Research Program of China (No. 2013CB338002).

References

1. Biehl, I., Meyer, B., Müller, V.: Differential fault attacks on elliptic curve cryptosystems. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 131–146. Springer, Heidelberg (2000)
2. Ciet, M., Joye, M.: Elliptic curve cryptosystems in the presence of permanent and transient faults. Des. Codes Cryptogr. **36**(1), 33–43 (2005)
3. Blömer, J., Otto, M., Seifert, J.-P.: Sign change fault attacks on elliptic curve cryptosystems. In: Breveglieri, L., Koren, I., Naccache, D., Seifert, J.-P. (eds.) FDTC 2006. LNCS, vol. 4236, pp. 36–52. Springer, Heidelberg (2006)
4. Schmidt, J., Medwed, M.: A fault attack on ECDSA. In: 2009 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 93–99. IEEE (2009)
5. Barenghi, A., Bertoni, G., Palomba, A., Susella, R.: A novel fault attack against ECDSA. In: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 161–166. IEEE (2011)
6. Nguyen, P.Q., Tibouchi, M.: Lattice-based fault attacks on signatures. Fault Analysis in Cryptography. ISC, pp. 201–220. Springer, Berlin (2012)
7. Howgrave-Graham, N., Smart, N.P.: Lattice attacks on digital signature schemes. Des. Codes Cryptogr. **23**(3), 283–290 (2001)

8. Nguyen, P.Q., Shparlinski, I.E.: The insecurity of the elliptic curve digital signature algorithm with partially known nonces. Des. Codes Cryptogr. **30**(2), 201–217 (2003)
9. Liu, M., Chen, J., Li, H.: Partially known nonces and fault injection attacks on SM2 signature algorithm. In: Lin, D., Xu, S., Yung, M. (eds.) Inscrypt 2013. LNCS, vol. 8567, pp. 343–358. Springer, Heidelberg (2014)
10. Leadbitter, P.J., Page, D.L., Smart, N.P.: Attacking DSA under a repeated bits assumption. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 428–440. Springer, Heidelberg (2004)
11. Faugère, J.-C., Goyet, C., Renault, G.: Attacking (EC)DSA given only an implicit hint. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 252–274. Springer, Heidelberg (2013)
12. Naccache, D., Nguyên, P.Q., Tunstall, M., Whelan, C.: Experimenting with faults, lattices and the DSA. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 16–28. Springer, Heidelberg (2005)
13. Office State Commercial Cryptgraphy Administration: Public key cryptographic algorithm SM2 based on elliptic curves (in Chinese) (2010). http://www.oscca.gov.cn/UpFile/2010122214822692.pdf
14. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. SPC. Springer, New York (2006)
15. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (1982)
16. Babai, L.: On lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
17. Stehlé, D., Albrecht, M., Cadé, D.: fplll-4.0 Lattice Reduction Library (2012). https://github.com/dstehle/fplll