Enhancing an Attack to DSA Schemes

Marios Adamoudis[1], Konstantinos A. Draziotis[2(✉)], and Dimitrios Poulakis[1]

[1] Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki, Greece
marios.p7@hotmail.com, poulakis@math.auth.gr
[2] Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
drazioti@csd.auth.gr

Abstract. In this paper, we improve the theoretical background of the attacks on the DSA schemes of a previous paper, and we present some new more practical attacks.

Keywords: Public key cryptography · Digital Signature Algorithm · Elliptic Curve Digital Signature Algorithm · Closest Vector Problem · LLL algorithm · BKZ algorithm · Babai's Nearest Plane Algorithm

MSC 2010: 94A60 · 11T71 · 11Y16

1 Introduction

In August 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed the Digital Signature Algorithm (DSA) for digital signatures [13,15]. This algorithm has become a standard [6] and was called Digital Signature Standard (DSS). In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized, see [10]. In the first subsection we recall the outlines of DSA and ECDSA.

1.1 The DSA and ECDSA Schemes

First, let us summarize DSA. The signer chooses a prime p of size between 1024 and 3072 bits with increments of 1024, as recommended in FIPS 186-3 [6, p. 15]. Also, he chooses a prime q of size 160, 224 or 256 bits, with $q|p-1$ and a generator g of the unique order q subgroup G of the multiplicative group $\mathbb{F}_p^*$ of the prime finite field $\mathbb{F}_p$. Furthermore, he selects a randomly $a \in \{1,\ldots,q-1\}$ and computes $R = g^a \mod p$. The public key of the signer is (p,q,g,R) and his private key a. He also publishes a hash function $h : \{0,1\}^* \to \{0,\ldots,q-1\}$. To sign a message $m \in \{0,1\}^*$, he selects randomly $k \in \{1,\ldots,q-1\}$ which is the ephemeral key, and computes $r = (g^k \mod p) \mod q$ and $s = k^{-1}(h(m) +$

M. Ćirić et al. (Eds.): CAI 2019, LNCS 11545, pp. 13–25, 2019.
https://doi.org/10.1007/978-3-030-21363-3_2

$ar) \bmod q$. The signature of m is the pair (r, s). The signature is valid if and only if we have:

$$r = ((g^{s^{-1}h(m) \bmod q} R^{s^{-1}r \bmod q}) \bmod p) \bmod q.$$

For the ECDSA the signer selects an elliptic curve E over $\mathbb{F}_p$, a point $P \in E(\mathbb{F}_p)$ with order a prime q of size at least 160 bits. According to FIPS 186-3, the prime p must be in the set $\{160, 224, 256, 512\}$. Further, he selects randomly $a \in \{1, \ldots, q-1\}$ and computes $Q = aP$. The public key of the signer is (E, p, q, P, Q) and his private key a. He also publishes a hash function $h : \{0,1\}^* \to \{0, \ldots, q-1\}$. To sign a message m, he selects randomly $k \in \{1, \ldots, q-1\}$ which is the ephemeral key and computes $kP = (x, y)$ (where x and y are regarded as integers between 0 and $p-1$). Next, he computes $r = x \bmod q$ and $s = k^{-1}(h(m) + ar) \bmod q$. The signature of m is (r, s). For its verification one computes

$$u_1 = s^{-1}h(m) \bmod q, \quad u_2 = s^{-1}r \bmod q, \quad u_1 P + u_2 Q = (x_0, y_0).$$

He accepts the signature if and only if $r = x_0 \bmod q$.

The security of the two systems is relied on the assumption that the only way to forge the signature is to recover either the secret key a, or the ephemeral key k (in this case is very easy to compute a). Thus, the parameters of these systems were chosen in such a way that the computation of discrete logarithms is computationally infeasible.

1.2 Our Contribution

Except the attacks in discrete logarithm problem, we have attacks based on the equality $s = k^{-1}(h(m) + ar) \bmod q$ which use lattice reduction techniques [1–5,11,12,16–19]. In this paper, we also use this equality and following the ideas of [19], we improve the efficiency of attacks on the DSA schemes described in it.

The attack described in [19] is based on a system of linear congruences of a particular form which has at most a unique solution below a certain bound, which can be computed efficiently. Thus, in case where the length of a vector, having as coordinates the secret and the ephemeral keys of some signed message is quite small, the secret key can be computed. More precisely in this work, we also consider the system of linear congruences of [19] and we improve the upper bound under which it has at most one solution. This extension provides an improvement of the attack [19], which also remains deterministic. Thus, when some signed messages are available, we can construct a such system whose solution has among its coordinates the secret key, and so it is possible to find it in practical time.

Furthermore, an heuristic improvement based on our attack is given. We update experimental results on (EC)DSA based on known bits of the ephemeral keys. In fact we prove that if we know 1 bit of a suitable multiple of the ephemeral keys for 206 signatures, we can find the secret key with success rate 62%. The previous best result was of Liu and Nguyen [12], where they provided a probabilistic attack based on enumeration techniques, where managed to find the

secret key if they know 2 bits of 100 ephemeral keys. The attack provided in [12] first reduces the problem of finding the secret key, to the hidden number problem (HNP) and then reduces HNP to a variant of CVP (called Bounded Decoded Distance problem : BDD).

The Structure of the Paper. The paper is organized as follows. In Sect. 2, we recall some basic results about lattices, and the Babai's Nearest Plane Algorithm. In Sect. 3, we prove some result which we need for the presentation of our attacks. Our attacks are presented in Sects. 4. Some experimental results are given in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Background on Lattices

In this section, we collect several well-known facts about lattices which form the background to our algorithms.

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ linearly independent vectors of $\mathbb{R}^m$. The set

$$\mathcal{L} = \left\{ \sum_{j=1}^{n} \alpha_j \mathbf{b}_j : \alpha_j \in \mathbb{Z}, 1 \leq j \leq n \right\}$$

is called a *lattice* and the set $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ a basis of $\mathcal{L}$. All the bases of $\mathcal{L}$ have the same number of elements n which is called *dimension* or *rank* of $\mathcal{L}$. If $n = m$, then the lattice $\mathcal{L}$ is said to have *full rank*. We denote by M the $n \times m$-matrix having as rows the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. If $\mathcal{L}$ has full rank, then the *volume* of the lattice $\mathcal{L}$ is defined to be the positive number $|\det M|$ which is independent from the basis $\mathcal{B}$. It is denoted by $vol(\mathcal{L})$ or $\det \mathcal{L}$ (see also [7]). If $\mathbf{v} \in \mathbb{R}^m$, then $\|\mathbf{v}\|$ denotes, as usually, the Euclidean norm of $\mathbf{v}$. We denote by $LLL(M)$, the application of well-known LLL-algorithm on the rows of M. Finally, we denote by $\lambda_1(\mathcal{L})$ the smaller of the lengths of vectors of $\mathcal{L}$.

We define the approximate Closest Vector Problem ($CVP_{\gamma_n}(L)$) as follows: Given a lattice $\mathcal{L} \subset \mathbb{Z}^m$ of rank n and a vector $\mathbf{t} \in \mathbb{R}^m$, find a vector $\mathbf{u} \in \mathcal{L}$ such that, for every $\mathbf{u}' \in \mathcal{L}$ we have:

$$\|\mathbf{u} - \mathbf{t}\| \leq \gamma_n \|\mathbf{u}' - \mathbf{t}\| \quad (\text{for some } \gamma_n \geq 1).$$

We say that we have a CVP oracle, if we have an efficient probabilistic algorithm that solves CVP_{γ_n} for $\gamma_n = 1$. To solve CVP_{γ_n}, we usually use Babai's algorithm [7, Chapter 18] (which has polynomial running time). In fact, combining this algorithm with LLL algorithm, we solve $\text{CVP}_{\gamma}(\mathcal{L})$ for some lattice $\mathcal{L} \subset \mathbb{Z}^m$, for $\gamma_n = 2^{n/2}$ and $n = rank(\mathcal{L})$ in polynomial time.

Babai's Nearest plane Algorithm:

`INPUT:` A $n \times m$-`matrix` M `with rows the vectors of a basis` $\mathcal{B} = \{\mathbf{b}_i\}_{1 \leq i \leq n} \subset \mathbb{Z}^m$ `of the lattice` $\mathcal{L}$ `and a vector` $\mathbf{t} \in \mathbb{R}^m$
`OUTPUT:` $\mathbf{x} \in L$ `such that` $||\mathbf{x} - \mathbf{t}|| \leq 2^{n/2} dist(L, \mathbf{t})$.

01. $M \leftarrow LLL(M)$ $(\delta = 3/4)$ # we can also use $BKZ_\beta(M)$
02. $M^* = \{(\mathbf{b}_j^*)_j\} \leftarrow GSO(M)$ # GSO : Gram-Schimdt Orthogonalization
03. $\mathbf{b} \leftarrow \mathbf{t}$
04. For $j = n$ to 1
05. $c_j \leftarrow \left\lfloor \frac{\mathbf{b} \cdot \mathbf{b}_j^*}{||\mathbf{b}_j^*||^2} \right\rceil$ $\#\lfloor x \rceil = \lfloor x + 0.5 \rfloor$
06. $\mathbf{b} \leftarrow \mathbf{b} - c_j \mathbf{b}_j$
07. Return $\mathbf{t} - \mathbf{b}$

Note that there is a variant of CVP, called BDD, where we search for vectors $\mathbf{u}$ such that $\|\mathbf{u} - \mathbf{t}\| \leq \lambda_1(L)/2$. Further, there are enumeration algorithms that compute all the lattice vectors within distance R from the target vector, see [8,9]. These algorithms are not of polynomial time with respect to the rank of the lattice.

3 Auxiliary Results

In this section we prove some results that we need for the description of our attack.

Proposition 1. *Let n, q and A_j be positive integers satisfying*

$$\frac{q^{\frac{j}{n+1}+f_q(n)}}{2} < A_j < \frac{q^{\frac{j}{n+1}+f_q(n)}}{1.5} \quad (j = 1, \ldots, n), \tag{1}$$

where $f_q(n)$ is a positive real number such that

$$f_q(n) + \frac{n}{n+1} < 1 \tag{2}$$

and

$$\frac{q^{1+2f_q(n)}}{1.5} < q - \frac{1}{2} q^{\frac{n}{n+1}+f_q(n)} \tag{3}$$

Let $\mathcal{L}$ be the lattice generated by the vectors

$$\mathbf{b}_0 = (-1, A_1, \ldots, A_n), \mathbf{b}_1 = (0, q, 0, \ldots, 0), \ldots, \mathbf{b}_n = (0, \ldots, 0, q).$$

Then, for all non-zero $\mathbf{v} \in \mathcal{L}$, we have:

$$\|\mathbf{v}\| > \frac{1}{2} q^{\frac{n}{n+1}+f_q(n)}.$$

Proof. Suppose that there is a vector $\mathbf{v} \in L \setminus \{\mathbf{0}\}$ such that

$$\|\mathbf{v}\| \leq \frac{1}{2} q^{\frac{n}{n+1}+f_q(n)}. \tag{4}$$

Then, the inequality (2) yields:

$$\|\mathbf{v}\| < \frac{q}{2} < q. \tag{5}$$

Since $\mathbf{v} \in L$, there are integers $x_0, x_1, \ldots, x_n$ such that

$$\mathbf{v} = x_0\mathbf{b}_0 + \cdots + x_n\mathbf{b}_n = (-x_0, x_0A_1 + x_1q, \ldots, x_0A_n + x_nq).$$

Thus, we deduce:

$$|x_0|, |x_0A_j + x_jq| \leq \frac{1}{2}q^{\frac{n}{n+1}+f_q(n)}.$$

If $x_0 = 0$, then we get the vector $\mathbf{v} = (0, x_1q, \ldots, x_nq)$ with length $> q$. On the other hand, (5) implies $\|\mathbf{v}\| < q$. Thus, we have a contradiction, and so we deduce that $x_0 \neq 0$.

Since $1 \leq |x_0| \leq \frac{1}{2}q^{n/(n+1)+f_q(n)}$, we distinguish the following two cases:

(i) There is $k \in \{1, 2, \ldots, n-1\}$ such that

$$q^{\frac{k-1}{n+1}} < |x_0| < q^{\frac{k}{n+1}}.$$

By (1), we obtain:

$$\frac{1}{2}\, q^{\frac{n+1-k}{n+1}+f_q(n)} \leq A_{n+1-k} \leq \frac{1}{1.5}\, q^{\frac{n+1-k}{n+1}+f_q(n)}.$$

Multiplying the two previous inequalities, we get:

$$\frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)} < |x_0|A_{n+1-k} < \frac{1}{1.5}\, q^{1+f_q(n)}.$$

By (3), we have:

$$\frac{q^{1+f_q(n)}}{1.5} < \frac{q^{1+2f_q(n)}}{1.5} < q - \frac{1}{2}q^{\frac{n}{n+1}+f_q(n)}.$$

It follows:

$$\frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)} < |x_0|A_{n+1-k} < q - \frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)} \tag{6}$$

If $x_0A_{n+1-k} + qx_{n+1-k} = 0$, then $|x_0|A_{n+1-k} = q|x_{n+1-k}| > q$, which contradicts the above inequality. Thus, we have $x_0A_{n+1-k} + qx_{n+1-k} \neq 0$.

Since $\|\mathbf{v}\| \geq |x_0A_{n+1-k} + qx_{n+1-k}|$, we get:

$$\|\mathbf{v}\| \geq ||x_0|A_{n+1-k} - q|x_{n+1-k}|| \geq q|x_{n+1-k}| - |x_0|A_{n+1-k}. \tag{7}$$

Assume that $x_{n+1-k} \neq 0$. It follows: $\|\mathbf{v}\| \geq q - |x_0|A_{n+1-k}$. Then, using the right part of inequality (6), we get:

$$\|\mathbf{v}\| > \frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)},$$

which contradicts (4). Hence, we have $x_{n+1-k} = 0$. By (7), we have: $\|\mathbf{v}\| \geq |x_0|A_{n+1-k}$. Using the left part of inequality (6) we obtain:

$$\|\mathbf{v}\| > \frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)},$$

which is a contradiction.

(ii) We have:

$$q^{\frac{n-1}{n+1}} < |x_0| < q^{\frac{n}{n+1}+f_q(n)}.$$

Further, (1) gives:

$$\frac{1}{2}\, q^{\frac{1}{n+1}+f_q(n)} \leq A_1 \leq \frac{1}{1.5}\, q^{\frac{1}{n+1}+f_q(n)}.$$

Multiplying the two inequalities we obtain:

$$\frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)} < |x_0|A_1 < \frac{1}{1.5}\, q^{1+2f_q(n)}.$$

By (3), we have:

$$\frac{1}{1.5}\, q^{1+2f_q(n)} < q - \frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)}.$$

Combining the two above inequalities, we deduce:

$$\frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)} < |x_0|A_1 < q - \frac{1}{2}\, q^{\frac{n}{n+1}+f_q(n)},$$

which is relation (6) with $k = n$. Proceeding, as previously, we obtain a contradiction. Thus, the result follows.

Remark 1. Proposition 1 is an improvement of [19, Lemma 1], since the obtained lower bound is better than the lower bound $\|\mathbf{v}\| > \frac{q^{n/(n+1)}}{8}$, obtained in [19, Lemma 1]. Furthermore, the number of signed messages in [19, Lemma 1], for q 160-bits, are less than $\ln \ln q \approx 4$. In the previous proposition the number of messages can be much larger. Note that a largest lower bound will allow us, as we shall see, to attack larger DSA keys (for fixed n and q).

Using the terminology of Proposition 1, we prove the following.

Proposition 2. *Let q and A_i, B_i $(i = 1, \ldots, n)$ be positive integers with A_i as in Proposition 1. Then, the system of congruences*

$$y_i + A_i x + B_i \equiv 0 \,(\mathrm{mod}\, q) \quad (i = 1, \ldots, n)$$

has at most one solution $\mathbf{v} = (x, y_1, \ldots, y_n)$ such that

$$\|\mathbf{v}\| < \frac{1}{4}\, q^{\frac{n}{n+1}+f_q(n)}.$$

If such $\mathbf{v}$ exists, then $\mathbf{v} = \mathbf{w} - \mathbf{b}$, where $\mathbf{b} = (0, B_1, \ldots, B_n)$, and $\mathbf{w}$ is a vector obtained by using a CVP oracle for the lattice $\mathcal{L}$ of Proposition 1 and $\mathbf{b}$.

Proof. Let $\mathbf{v} = (x, y_1, \ldots, y_n)$ be a solution of the system with

$$\|\mathbf{v}\| < \frac{1}{4}\, q^{\frac{n}{n+1}+f_q(n)}.$$

We denote by $\mathcal{L}$ the lattice spanned by the rows of the $(n+1)\times(n+1)$ matrix

$$\begin{bmatrix} -1 & A_1 & A_2 & \dots & A_n \\ 0 & q & 0 & \dots & 0 \\ 0 & 0 & q & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & q \end{bmatrix} \tag{8}$$

and set $\mathbf{b} = (0, B_1, \dots, B_n)$. Since $y_i + A_i x + B_i \equiv 0 \,(\mathrm{mod}\, q)$ there is $z_i \in \mathbb{Z}$, such that $y_i + B_i = -A_i x + z_i q$. Set $\mathbf{u} = \mathbf{v} + \mathbf{b} = (x, y_1 + B_1, \dots, y_n + B_n)$. Then $\mathbf{u} = (x, -A_1 x + z_1 q, \dots, -A_n x + z_n q)$ belongs to $\mathcal{L}$ and we have

$$\|\mathbf{u} - \mathbf{b}\| = \|\mathbf{v}\| < \frac{1}{4}\, q^{\frac{n}{n+1} + f_q(n)}.$$

On the other hand, using the CVP-oracle, we compute $\mathbf{w} \in \mathcal{L}$ such that

$$\|\mathbf{w} - \mathbf{b}\| \le \|\mathbf{u} - \mathbf{b}\| < \frac{1}{4}\, q^{\frac{n}{n+1} + f_q(n)}.$$

Thus, we get: $\|\mathbf{w} - \mathbf{u}\| \le \|\mathbf{w} - \mathbf{b}\| + \|\mathbf{b} - \mathbf{u}\| < \frac{1}{2}\, q^{\frac{n}{n+1} + f_q(n)}$. Since $\mathbf{w} - \mathbf{u} \in \mathcal{L}$, Proposition 1 implies $\mathbf{w} = \mathbf{u}$. So, the CVP oracle outputs the vector $\mathbf{u}$ and so we can compute $\mathbf{v} = \mathbf{u} - \mathbf{b}$.

To get an idea how to choose the sequence $f_q(n)$ we use the following well known result.

Lemma 1. *(Hermite). For every full rank lattice $\mathcal{L} \subset \mathbb{R}^n$ we have:*

$$\lambda_1(\mathcal{L}) \le \sqrt{n}\,(\det \mathcal{L})^{1/n}.$$

In our case we get $\lambda_1(L) \le \sqrt{n+1}\, q^{\frac{n}{n+1}}$. The lower bound of Proposition 1 is of the form $\frac{1}{2} q^x$. So, we get:

$$\frac{1}{2} q^x \le \lambda_1(L) \le \sqrt{n+1}\, q^{\frac{n}{n+1}}.$$

Solving with respect to x, we obtain:

$$x \le \frac{\ln 2 + \frac{1}{2}\ln(n+1)}{\ln q} + \frac{n}{n+1} = \frac{\ln(4(n+1))}{2\ln q} + \frac{n}{n+1}.$$

Thus, we get an upper bound for x. For instance we may select

$$f_q(n) = \frac{\ln 2 + \ln(n+1)}{2n \ln q}.$$

In general, we can choose

$$f_q(n) = g_{q,b,c,d}(n) = \frac{c \ln(n+1)}{b n^d \ln q}, \tag{9}$$

where b, c, d are chosen such that $0 < g_{q,b,c,d}(n) < \frac{\ln(4(n+1))}{2\ln q}$.

4 The Attack

In this section we describe our attack. Let m_i $(i = 1, \ldots, n)$ be messages signed with (EC)DSA system and (r_i, s_i) their signatures. Then, there are $k_i \in \{1, \ldots, q-1\}$ such that $r_i = (g^{k_i} \bmod p) \bmod q$ (resp. $r_i = x_i \bmod q$ and $k_i P = (x_i, y_i)$) and $s_i = k_i^{-1}(h(m_i) + ar_i) \bmod q$. It follows that

$$k_i + C_i a + D_i \equiv 0 \pmod{q} \quad (i = 1, \ldots, n),$$

where $C_i = -r_i s_i^{-1} \bmod q$ and $D_i = -s_i^{-1} h(m_i) \bmod q$. Multiplying both sides by $C_i^{-1} \bmod q$, we get: $C_i^{-1} k_i + a + C_i^{-1} D_i \equiv 0 \,(\mathrm{mod}\, q)$.

Now, we pick integers A_i satisfying

$$\frac{q^{\frac{i}{n+1}+f_q(n)}}{2} < A_i < \frac{q^{\frac{i}{n+1}+f_q(n)}}{1.5}$$

and we multiply by A_i both sides of the above congruence. So, we get:

$$A_i C_i^{-1} k_i + A_i a + A_i C_i^{-1} D_i \equiv 0 \,(\mathrm{mod}\, q).$$

Set $B_i = A_i C_i^{-1} D_i \bmod q$ $(i = 1, \ldots, n)$. Then, the vector

$$\mathbf{s} = (a, A_1 C_1^{-1} k_1 \bmod q, \ldots, A_n C_n^{-1} k_n \bmod q)$$

satisfies the system

$$y_i + A_i x + B_i \equiv 0 \,(\mathrm{mod}\, q) \quad (i = 1, \ldots, n). \tag{10}$$

We set $\mathbf{b} = (0, B_1, \ldots, B_n)$ and $M_{n,q} = \frac{1}{4} q^{\frac{n}{n+1}+f_q(n)}$, for some sequence $f_q(n)$ satisfying the hypotheses of Proposition 1. The vectors

$$\mathbf{b}_0 = (-1, A_1, \ldots, A_n), \quad \mathbf{b}_1 = (0, q, 0, \ldots, 0), \ldots, \mathbf{b}_n = (0, \ldots, 0, q),$$

form a basis of $\mathbb{R}^{n+1}$. We denote by $\mathcal{L}$ the lattice spanned by $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n$. If $\|\mathbf{s}\| < M_{n,q}$, then Proposition 2 implies that $\mathbf{s} = \mathbf{w} - \mathbf{b}$, where $\mathbf{w}$ is a vector obtained by using a CVP oracle for $\mathcal{L}$ and $\mathbf{b}$. Thus, we can compute the secret key a which is the first coordinate of $\mathbf{s}$.

An Improvement of the Attack. The previous attack, which is deterministic, needs quite short solution vector (and so small secret key) in order to succeed. Now, we remark that if $\mathbf{u} \in \mathcal{L}$, then $\mathbf{u} - \mathbf{b}$ is a solution of the system (10). Indeed, since $\mathbf{u} \in \mathcal{L}$, there are integers $l_0, \ldots, l_n$ such that $\mathbf{u} = l_0 \mathbf{b_0} + \cdots + l_n \mathbf{b_n}$, and so we get:

$$\mathbf{u} - \mathbf{b} = (-l_0, l_0 A_1 + l_1 q - B_1, \ldots, l_0 A_n + l_n q - B_n) = (x, y_1, \ldots, y_n).$$

Thus, we obtain that $y_i + A_i x + B_i = l_i q \equiv 0 \pmod{q}$. So, the vectors of the form $\mathbf{x} - \mathbf{b}$, where $\mathbf{x}$ is in the set

$$\mathcal{A}_{\gamma_n}(\mathbf{b}) = \{\mathbf{u} \in L : ||\mathbf{u} - \mathbf{b}|| < \gamma_n M_{n,q}\},$$

are solutions of the system (10), for some positive number γ_n.

The parameters we usually use in a (EC)DSA system, provides a solution $\mathbf{s} = (x, y_1, \ldots, y_n)$, where all the entries are 160-bits integers and smaller than q. In this case, we compute experimentally that the value of $\gamma_n = \frac{||\mathbf{s}||}{M_{n,q}}$ is on average approximately equal to 38. For larger values of γ_n, Babai's algorithm (or some other approximation algorithm) may succeed in finding in polynomial time such a vector. Indeed, we checked (see Sect. 5) that Babai's algorithm managed to find a solution for $\gamma_n \in [44, 52]$. We get such values of γ_n in the case where we have smaller keys. We provide the details of these experiments in Sect. 5.

This attack is non deterministic, since $\gamma_n > 1$ and so Proposition 2 does not hold. If $\gamma_n = 1$, then Proposition 2 holds, and the attack is deterministic. Below we present our attack.

Babai's Attack

Input : A public key (p, q, g, R) of a DSA scheme or a public key (E, p, q, P, Q) of a ECDSA scheme. Further, m signed messages are given.
Output : The secret key or Fail.
1. Choose $f_q(n)$.
2. Pick the maximum integer $n > 0$ such that there is an integer in the intervals

$$I_i = \left(\frac{q^{i/(n+1)+f_q(n)}}{2}, \frac{q^{i/(n+1)+f_q(n)}}{1.5} \right), \quad 1 \leq i \leq n.$$

3. If $n \leq m$, then go to the next step, else return fail.
4. Choose randomly A_i from I_i.
5. Set $\mathbf{b} = (0, B_1, \ldots, B_n)$ and construct the system

$$y_i + A_i x + B_i \equiv 0 \pmod{q} \quad (i = 1, \ldots, n).$$

6. Construct the lattice $\mathcal{L}$, generated by the rows of the DSA matrix (8).
7. Compute $B = LLL(A)$.
8. Apply Babai's Nearest Plane Algorithm in the rows of matrix B with target vector $\mathbf{b}$. Let $\mathbf{s}$ be the output.
9. If the first coordinate s_1 of $\mathbf{s}$ satisfy $g^{s_1} = R$, (respectively $Q = s_1 P$) in $\mathbb{F}_p^*$, return s_1, else return fail.

The attacker, say Eve, has to make the choice of n and f_q. The choice of n is not random. A minimal condition is to choose it, in such a way that the interval I_i contains at least one integer. Then, she can construct a system of the form $y_i + A_i x + B_i \equiv 0 \pmod{q}$, $1 \leq i \leq n$, and $A_i \in I_i$. So, in practice the difficult part for Eve, is to find n signed messages. So Eve must be an active attacker and uses the DSA system as a signing oracle.

Remark 2. (i) Proposition 2 may not be satisfied, but the attack may return the secret key. This is because the hypotheses of Proposition 2 are only necessary and not sufficient.

(ii) The attack may fail in the sense that will not compute the secret key. The probability of success depends on the choices of A_i.

(iii) In the 7th step we can use BKZ algorithm with a suitable blocksize instead of LLL. This is something common in case we want a more reduced basis.

5 Experimental Results

For the following experiments we used the computer algebra system Sagemath [20][1]. We generated 100 DSA-systems of the form (10), $y_i + A_i x + B_i \equiv 0 \pmod{q}$, $1 \leq i \leq n$, having a solution $\mathbf{s} = (x, y_1, \ldots, y_n)$ where x is the secret key. Once q is fixed to a 160 bit prime, we choose a natural number $n = n_0$ such that the intervals I_i contain at least one integer and a sequence $f_q(n)$, such that the value $f_q(n_0)$ satisfies the inequalities (2) and (3) of Proposition 1. Then, we choose the secret key x and y_i (we call y_i's *derivative ephemeral keys*), such that Proposition 2 is satisfied. In fact we pick the solution vector $\mathbf{s}$ such that

$$\frac{q^{n/(n+1)}}{16} < \|\mathbf{s}\| < \frac{q^{n/(n+1)+f_q(n)}}{4}.$$

The attack in [19], it may fail for the previous experiments, since the norm of solution we are looking for, is larger than $\frac{q^{n/(n+1)}}{16}$ and $n \gg \ln(\ln(q))$. But our attack will succeed since Proposition 2 is satisfied. The attack is deterministic. A choice that satisfies the previous is q : a 160-bit prime, $n_0 = 14$, $f_q(n) = \frac{\ln(n+1)}{n \ln q}$. We generated 100 DSA systems with the previous parameters and secret key 147 bits and derivative ephemeral keys 145 bits. As a CVP oracle, we used Babai algorithm. In all the instances we found the secret key, as our attack suggests. We tested our improvement (Babai's attack), which is no longer deterministic. We summarize the results in Table 1.

Remark 3. In the Example studied in [19] some typographic errors occurred in the values of the quantities A_1, $h(m_2)$, $h(m_3)$, s_2 and s_3. The correct values are $A_1 = 32D_1$ (and so, we have $l_1 = a^{-1}k_1 32 \mod q < 2^{96}$), and

$$\begin{aligned}
h(m_2) &= 432847687632257989627045945667165545993050789339,\\
h(m_3) &= 102247883422181353858596598828981363231626289233,\\
s_2 &= 128664406831208422446798919343676926547176728457 1,\\
s_3 &= 135723554005178129314372023275275184067724775409 0.
\end{aligned}$$

[1] The code can be found in https://github.com/drazioti/python_scripts/tree/master/paper_dsa.

Table 1. We set $n = 206$ and $f_q(n) = \frac{170 \ln (n+1)}{n \ln q}$. We generated 100 random DSA systems for each row. The pair (α, β) at the first column, means that we pick the secret key uniformly from $[2^{\alpha-1}, 2^{\alpha} - 1]$ and the derivative ephemeral keys from $[2^{\beta-1}, 2^{\beta} - 1]$ (and fixed 160 bit prime q). For preprocessing we used BKZ with blocksize 70. The second column is the average value of γ_n. The last column contains the percentage that Babai's attack succeeds in finding the solution i.e. the secret key.

bits:(Skey, Der.Ep.keys)	γ_n	suc.rate
$(158, 157)$	43.81	17%
$(158, 155)$	51.34	100%
$(157, 157)$	44.15	23.3%
$(157, 156)$	48.68	100%

A Further Heuristic Improvement. We can further improve the previous results. The idea is to use another target vector instead of $\mathbf{b} = (0, B_1, \ldots, B_n)$. We consider the following vector

$$\mathbf{b} = (\varepsilon, \varepsilon + B_1, \ldots, \varepsilon + B_n),$$

where $\varepsilon = 2^{159} - 2^{157}$. That is the new target vector is equal with the previous plus the vector $(\varepsilon, \ldots, \varepsilon)$. The reason for picking such a target vector, is because the entries of the solution vector are balanced, i.e. of the same bit lengths. We test the previous idea, by choosing the solution vectors

$$\mathbf{s} \in \{2^{159}, \ldots, 2^{160} - 1\} \times \{2^{158}, \ldots, 2^{159} - 1\}^n.$$

That is the secret key has 160−bits and the derivative ephemeral keys 159−bits. We considered 100 DSA such systems with $n = 206$ and $f_q(n)$ as in Table 1, preprocessing BKZ-85, and our algorithm found the secret keys in 62 instances. The time execution per example was on average 2 min in an I3 Intel CPU. So, having one least (or most) significant bit for 206 derivative ephemeral keys, we can find the secret key. This result improves, in some sense, the result [12, Sect. 4.4], where with 100 signatures and knowing 2 least significant bits of the ephemeral keys, they computed the secret key with success rate 23% and in 4185 s on average per instance.

6 Conclusion

Following the ideas of [19], we have presented new attacks on DSA schemes. First, we have improved the bound for the lengths of the secret and ephemeral keys and we replaced the use of Micciancio-Voulgaris algorithm [14], by the polynomial complexity Babai's algorithm. This allowed us to find larger keys than in [19]. Our attack remained deterministic. Furthermore, we presented an heuristic extension of our attack for finding even larger secret keys. Finally, we improved the state of the art attack in (EC)DSA presented in [12]. Several experiments were described.

References

1. Bellare, M., Goldwasser, S., Micciancio, D.: "Pseudo-random" number generation within cryptographic algorithms: the DDS case. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 277–291. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052242
2. Blake, I.F., Garefalakis, T.: On the security of the digital signature algorithm. Des. Codes Cryptogr. **26**(1–3), 87–96 (2002)
3. Draziotis, K.A., Poulakis, D.: Lattice attacks on DSA schemes based on Lagrange's algorithm. In: Muntean, T., Poulakis, D., Rolland, R. (eds.) CAI 2013. LNCS, vol. 8080, pp. 119–131. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40663-8_13
4. Draziotis, K.A.: (EC)DSA lattice attacks based on Coppersmith's method. Inform. Proc. Lett. **116**(8), 541–545 (2016)
5. Faugère, J.-L., 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). https://doi.org/10.1007/978-3-642-35999-6_17
6. FIPS PUB 186–3, Federal Information Processing Standards Publication, Digital Signature Standard (DSS)
7. Galbraith, S.: Mathematics of Public Key Cryptography. Cambridge University Press, Cambridge (2012)
8. Hanrot, G., Pujol, X., Stehlé, D.: Algorithms for the shortest and closest lattice vector problems. In: Chee, Y.M., et al. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 159–190. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20901-7_10
9. Hanrot, G., Stehlé, D.: Improved analysis of Kannan's shortest lattice vector algorithm. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 170–186. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_10
10. Johnson, D., Menezes, A.J., Vanstone, S.A.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**, 36–63 (2001)
11. Howgrave-Graham, N.A., Smart, N.P.: Lattice attacks on digital signature schemes. Des. Codes Cryptogr. **23**, 283–290 (2001)
12. Liu, M., Nguyen, P.Q.: Solving BDD by enumeration: an update. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 293–309. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36095-4_19
13. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
14. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In: Proceedings of the 42nd ACM Symposium on Theory of Computing - STOC 2010, pp. 351–358. ACM (2010)
15. National Institute of Standards and Technology (NIST). FIPS Publication 186: Digital Signature Standard, May 1994
16. Nguyen, P.Q., Shparlinski, I.E.: The insecurity of the digital signature algorithm with partially known nonces. J. Cryptology **15**, 151–176 (2002)
17. Nguyen, P.Q., Shparlinski, I.E.: The insecurity of the elliptic curve digital signature algorithm with partially known nonces. Des. Codes Cryptogr. **30**, 201–217 (2003)

18. Poulakis, D.: Some lattice attacks on DSA and ECDSA. Appl. Algebra Eng. Commun. Comput. **22**, 347–358 (2011)
19. Poulakis, D.: New lattice attacks on DSA schemes. J. Math. Cryptol. **10**(2), 135–144 (2016)
20. Sage Mathematics Software, The Sage Development Team (version 8.1). http://www.sagemath.org