The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces

Phong Q. Nguyen*
Département d'Informatique, École Normale Supérieure,
45, rue d'Ulm, 75230 Paris Cedex 05, France
Email: pnguyen@ens.fr
URL: http://www.di.ens.fr/~pnguyen

and

Igor E. Shparlinski†
Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
Email: igor@comp.mq.edu.au
URL: http://www.comp.mq.edu.au/~igor/

Abstract. Nguyen and Shparlinski have recently presented a polynomial-time algorithm that provably recovers the signer's secret DSA key when a few consecutive bits of the random nonces k (used at each signature generation) are known for a number of DSA signatures at most linear in $\log q$ (q denoting as usual the small prime of DSA), under a reasonable assumption on the hash function used in DSA. The number of required bits is about $\log^{1/2} q$, but can be decreased to $\log \log q$ with a running time $q^{O(1/\log \log q)}$ subexponential in $\log q$, and even further to 2 in polynomial time if one assumes access to ideal lattice basis reduction, namely an oracle for the lattice closest vector problem for the infinity norm. All previously known results were only heuristic, including those of Howgrave-Graham and Smart who introduced the topic. Here, we obtain similar results for the elliptic curve variant of DSA (ECDSA).

Keywords: Cryptanalysis, ECDSA, lattices, LLL, closest vector problem, distribution, discrepancy, exponential sums, elliptic curves.

1. Introduction

* Work supported in part by the RNRT "Turbo-signatures" project of the French Ministry of Research.
† Supported in part by ARC.

To appear in the *Design, Codes and Cryptography.*

1.1. The Digital Signature Algorithm (DSA)

Recall the *Digital Signature Algorithm* (see [20, 30]), or DSA, used in the American federal digital signature standard [22].

Let p and $q \geq 3$ be prime numbers with $q|p-1$. As usual $\mathbb{F}_p$ and $\mathbb{F}_q$ denote fields of p and q elements which we assume to be represented by the elements $\{0, \ldots, p-1\}$ and $\{0, \ldots, q-1\}$ respectively. For integers s and $m \geq 1$ we denote by $\lfloor s \rfloor_m$ the remainder of s on division by m. We also use $\log z$ to denote the binary logarithm of $z > 0$.

Let $\mathcal{M}$ be the set of messages to be signed and let $h : \mathcal{M} \to \mathbb{F}_q$ be an arbitrary hash-function. The signer's secret key is an element $\alpha \in \mathbb{F}_q^*$.

Let $g \in \mathbb{F}_p$ be a fixed element of multiplicative order q, that is $g^q = 1$ and $q \neq 1$ which is *publicly* known. To sign a message $\mu \in \mathcal{M}$, one chooses a random integer $k \in \mathbb{F}_q^*$ usually called the *nonce*, and which must be kept secret. One then defines the following two elements of $\mathbb{F}_q$:

$$\begin{aligned} \rho(k) &= \left\lfloor \left\lfloor g^k \right\rfloor_p \right\rfloor_q \\ \sigma(k,\mu) &= \left\lfloor k^{-1} \left(h(\mu) + \alpha\rho(k) \right) \right\rfloor_q . \end{aligned}$$

The pair $(\rho(k), \sigma(k,\mu))$ is the *DSA signature* of the message μ with a nonce k. In practice, q is usually of bit-length 160 and p is of bit-length between 512 and 1024.

1.2. The Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of DSA (see [13, 14, 15]). ECDSA uses an elliptic curve $\mathbb{E}$ over $\mathbb{F}_p$ where p is prime. If $N = |\mathbb{E}(\mathbb{F}_p)|$ denotes the number of rational points over $\mathbb{F}_p$, it is well-known that

$$|N - p - 1| \leq 2p^{1/2}$$

and that $\mathbb{E}(\mathbb{F}_p)$ together with the point at infinity $\mathcal{O}$ form an Abelian group, see [29].

Let $G \in \mathbb{E}(\mathbb{F}_p)$ be a fixed point of order q, where q is a prime divisor of N, that is $qG = \mathcal{O}$, where $\mathcal{O}$ is the point at infinity. Both G and $q \neq 1$ are *publicly* known. For a point $Q \in \mathbb{E}(\mathbb{F}_p)$ we denote by $x(Q)$, $0 \leq x(Q) \leq p-1$, the first component of $Q = (x, y)$ in the affine model of $\mathbb{E}$. The signer's secret key is again an element $\alpha \in \mathbb{F}_q^*$.

To sign a message $\mu \in \mathcal{M}$, one chooses a random integer $k \in \mathbb{F}_q^*$ usually called the *nonce*, and which must be kept secret. One then defines the following two elements of $\mathbb{F}_q$:

$$\begin{aligned} r(k) &= \lfloor x(kG) \rfloor_q \\ s(k,\mu) &= \left\lfloor k^{-1} \left(h(\mu) + \alpha r(k) \right) \right\rfloor_q \end{aligned}$$

The pair $(r(k), s(k,\mu))$ is the *ECDSA signature* of the message μ with a nonce k. As in the case of DSA, in practice, q is usually of bit-length around 160.

1.3. Former results

In a certain computational model, the security of ECDSA (with respect to adaptive chosen-message attacks) can be proved, see [7]. However, serious precautions must be taken with the generation of the nonce k, as most results applying to DSA can be extended to ECDSA. It is well-known that if k is disclosed, then the secret key α can easily be recovered. It has been shown by Bellare *et al.* [3] that one can still recover α if the nonce k is produced by Knuth's linear congruential generator with known parameters, or variants. That attack is provable under the random oracle model, and relies on Babai's approximation algorithm [2] for the closest vector problem (CVP) in a lattice, which is based on the celebrated LLL algorithm [19]. The attack does not work if the parameters of the generator are unknown.

Recently, Howgrave-Graham and Smart [12] introduced a different scenario to study the security of DSA. Suppose that for a reasonable number of signatures, a small fraction of the corresponding nonce k is revealed. For instance, suppose that the ℓ least significant bits of k are known. Howgrave-Graham and Smart proposed in [12] several heuristic attacks to recover the secret key in such setting and variants (known bits in the middle, or split in several blocks) when ℓ is not too small. Like [3], the attacks make use of LLL-based Babai's CVP approximation algorithm [2]. However, the attacks of [3] and [12] are quite different. Howgrave-Graham and Smart have followed an applied approach. The attack used several heuristic assumptions which did not allow precise statements on its theoretical behaviour. It has been assumed that the DSA signatures followed a perfectly uniform distribution, that some lattice enjoyed some natural however heuristic property, and that Babai's algorithm behaves much better than theoretically guaranteed. Some heuristic arguments of this attack have been sharpened by Nguyen [23].

Nguyen and Shparlinski [24], following the approach of [23], have improved the analysis of the attack of Howgrave-Graham and Smart [12],

using the work of Boneh and Venkatesan on the hardness of Diffie-Hellman bits [5]. They showed that there is a *provable polynomial-time attack* against DSA when the nonces are partially known, under two reasonable assumptions: the size of q should not be too small compared to p, and the probability of collisions for the hash function h should not be too large compared to $1/q$. More precisely, under such conditions, if for a certain (polynomially bounded) number of random messages $\mu \in \mathcal{M}$ and random nonces $k \in [1, q-1]$ about $\log^{1/2} q$ least significant bits of k are known, then in polynomial time one can recover the signer's secret key α. The same result holds for the most significant bits when one uses an appropriate definition of the most significant bits tailored to modular residues. With the usual definition of most significant bits, one needs one more bit than in the case of least significant bits, as q might be only marginally larger than a power of two (in which case the amount of information is less than for least significant bits). The result is slightly worse for arbitrary windows of consecutive bits: in such a case, one requires twice as many bits. For least significant bits (or appropriate most significant bits), the number of bits can be decreased to 2 if one further assumes access to ideal lattice reduction (namely, an oracle for the closest vector problem for the infinity norm). Such an assumption is realistic in low dimension despite NP-hardness results on lattice problems, due to the well-known experimental fact that state-of-the-art lattice basis reduction algorithms behave much better than theoretically guaranteed. Alternatively, the number of bits can be decreased to $\log \log q$ but with a running time $q^{O(1/\log \log q)}$ subexponential in $\log q$, using the closest vector approximation algorithm of [1, Corollary 16]. This running time is interesting, as the bit-length of q is usually chosen to be 160, in order to avoid square-root attacks.

1.4. Our results

In this paper, we extend the results of Nguyen and Shparlinski [24] on DSA to the case of ECDSA. This provides the first provable polynomial-time attack against ECDSA when the nonces are partially known, under the same two reasonable assumptions. Although the results previously mentioned on DSA could heuristically be applied to ECDSA, no proved result has been known, due to the potential difference of distribution between the ECDSA signatures $(r(k), s(k, \mu))$ and the DSA signatures $(\rho(k), \sigma(k, \mu))$.

In fact, our approach is very similar to that of [24], the main difference being that we use bounds of exponential sums from [16] to obtain some results on the distribution of ECDSA signatures $(r(k), s(k, \mu))$, whereas [24] applies bounds of exponential sums from [17] to study

the distribution of DSA signatures $(\rho(k), \sigma(k,\mu))$ Accordingly, because of the difference in the strength of the bounds of exponential sums from [16] and [17], our results on the ECDSA are slightly weaker that those of [24] for DSA.

The attack of [24] works in practice when 3 bits are leaked for about a hundred signatures. The situation is the same with ECDSA. However, those experimental results are superseded by the recent important result of Bleichenbacher [4]. Bleichenbacher presented a heuristic attack not based on lattices against DSA when some of the bits of the nonces are known. Currently, the best experimental result with this attack is that one can recover the secret key given a leakage of $\log 3 \approx 1.58$ bits for 2^{22} signatures, in about 3 days on a 450 MHz Ultrasparc using 500Mb of RAM. Naturally, this heuristic attack also applies to the case of ECDSA. However our method is the only one yielding provable results at the moment.

1.5. Overview of our attack

Like [24], our attack follows the approach of Nguyen [23] which reduces the DSA/ECDSA problem to a variant of the *hidden number problem* (HNP) introduced in 1996 by Boneh and Venkatesan [5, 6]. The HNP can be stated as follows: recover a number $\alpha \in \mathbb{F}_q$ such that for many known random $t \in \mathbb{F}_q$, an approximation $\mathrm{APP}_{\ell,q}(\alpha t)$ of αt is known. Here, for any rationals n and ℓ, the notation $\mathrm{APP}_{\ell,q}(n)$ denotes any rational r such that:

$$|n - r|_q \leq \frac{q}{2^{\ell+1}},$$

where the symbol $|.|_q$ is defined as $|z|_q = \min_{b\in\mathbb{Z}} |z - bq|$ for any real z.

The connection between the ECDSA problem and HNP can easily be explained. Assume that we know the ℓ least significant bits of a nonce $k \in \mathbb{F}_q^*$. That is, we are given an integer a such that $0 \leq a \leq 2^\ell - 1$ and $k - a = 2^\ell b$ for some integer $b \geq 0$. Given a message μ signed with the nonce k, the congruence

$$\alpha r(k) \equiv s(k,\mu)k - h(\mu) \pmod{q},$$

can be rewritten for $s(k,\mu) \neq 0$ as:

$$\alpha r(k) 2^{-\ell} s(k,\mu)^{-1} \equiv \left(a - s(k,\mu)^{-1} h(\mu)\right) 2^{-\ell} + b \pmod{q}. \qquad (1)$$

Now define the following two elements

$$\begin{aligned} t(k,\mu) &= \left\lfloor 2^{-\ell} r(k) s(k,\mu)^{-1} \right\rfloor_q \\ u(k,\mu) &= \left\lfloor 2^{-\ell} \left(a - s(k,\mu)^{-1} h(\mu)\right) \right\rfloor_q \end{aligned}$$

and remark that both $t(k,\mu)$ and $u(k,\mu)$ can easily be computed by the attacker from the publicly known information. Recalling that $0 \leq b \leq q/2^\ell$, we obtain

$$0 \leq \lfloor \alpha t(k,\mu) - u(k,\mu) \rfloor_q < q/2^\ell.$$

And therefore:

$$|\alpha t(k,\mu) - u(k,\mu) - q/2^{\ell+1}|_q \leq q/2^{\ell+1}. \tag{2}$$

Thus, an approximation $\mathrm{APP}_{\ell,q}(\alpha t(k,\mu))$ is known. Collecting several relations of this kind for several pairs (k,μ), the problem of recovering the secret key α is thus a HNP in which the distribution of the multiplier $t(k,\mu)$ is not necessarily perfectly uniform, and which at first sight seems hard to study. This problem of recovering will be called ECDSA–HNP in the rest of the paper.

To solve ECDSA–HNP, we apply a lattice-based algorithm proposed by Boneh and Venkatesan in [5], which relies on a simple reduction from HNP to CVP. This polynomial-time algorithm, which we will call BV, is again based on Babai's CVP approximation algorithm. It provably solves HNP when $\ell \geq \log^{1/2} q + \log\log q$. That result enabled Boneh and Venkatesan to establish in [5] some results on the bit-security of the Diffie–Hellman key exchange and related cryptographic schemes. However, in the latter application, the distribution of the multipliers t is not perfectly uniform, making some of the statements of [5] incorrect. This has led Gonzáles Vasco and Shparlinski [10] to extend results on the BV algorithm to the case where t is randomly selected from a subgroup of $\mathbb{F}_q^*$, to obtain rigorous statements on the bit-security of the Diffie–Hellman key exchange and related schemes (see also [11]).

In ECDSA–HNP as well, the distribution of the multiplier $t(k,\mu)$ is not necessarily perfectly uniform. We thus apply an extension (presented in [24]) of the results of [5] on the BV algorithm using the notion of discrepancy, in the spirit of that of [10, 11]. To achieve the proof of our attack, we show using exponential sum techniques that ECDSA signatures follow some kind of uniform distribution. A similar reasoning has been exploited in [24].

1.6. Structure of the paper and notation

The paper is organized as follows. In Section 2, we review a few facts on the hidden number problem, and we recall two extensions of [5, Theorem 1] where the multipliers may have imperfect uniform distribution. In Section 3, we obtain uniformity results on the distribution of ECDSA signatures, which might be of independent interest. Finally, in Section 4, we collect the aforementioned results and apply it to ECDSA.

Throughout the paper the implied constants in symbols 'O' may occasionally, where obvious, depend on the small positive parameter ε and are absolute otherwise; they all are effective and can be explicitly evaluated.

We use $[\alpha, \beta]$ and $]\alpha, \beta[$ to denote the closed and open intervals, respectively.

As usual, $\Pr(\mathcal{E})$ denotes the probability of an event $\mathcal{E}$.

For a real x, $\lfloor x \rfloor$ denotes the integer part of x, that is the integer n such that $n \le x < n+1$. $\lceil x \rceil$ is the integer n such that $n \ge x > n-1$.

2. The Hidden Number Problem

In their analysis of the hidden number problem, Boneh and Venkatesan [5] presented the following lattice-based result. Let α be some integer $[1, q-1]$ and $n = \lceil \log q \rceil$. Let $\mathcal{O}$ be a function defined by $\mathcal{O}(t) = \mathrm{MSB}_{\ell,q}(\alpha t \bmod q)$ with $\ell = \lceil n^{1/2} \rceil + \lceil \log n \rceil$. Then [5, Theorem 1] states that there exists a deterministic polynomial time algorithm $\mathcal{A}$ such that

$$\Pr_{t_1,\ldots,t_d} \left[\mathcal{A}\left(t_1, \ldots, t_d, \mathcal{O}(t_1), \ldots, \mathcal{O}(t_d)\right) = \alpha \right] \ge \frac{1}{2},$$

where $d = 2\lceil n^{1/2} \rceil$ and $t_1, \ldots, t_d$ are chosen uniformly and independently at random from $\mathbb{Z}_q^*$.

To show the insecurity of DSA with partially known nonces, Nguyen and Shparlinski [24] generalized the previous result to cases where the multiplier t has not necessarily perfectly uniform distribution. The generalization used the classical notion of discrepancy [8, 18, 28]. Recall that the *discrepancy* $\mathcal{D}(\Gamma)$ of an N-element sequence $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$ of elements of the interval $[0, 1]$ is defined as

$$\mathcal{D}(\Gamma) = \sup_{J \subseteq [0,1]} \left| \frac{A(J, N)}{N} - |J| \right|,$$

where the supremum is extended over all subintervals J of $[0, 1]$, $|J|$ is the length of J, and $A(J, N)$ denotes the number of points γ_n in J for $0 \le n \le N-1$.

Informally speaking the discrepancy tells us how much the number of hits $A(J, N)$ of a given interval J differ from its expected value $|J|N$.

Nguyen and Shparlinski [24] introduced the following definition: a finite sequence $\mathcal{T}$ of integers is *Δ-homogeneously distributed modulo q* if for any integer a coprime with q, the discrepancy of the sequence $\{\lfloor at \rfloor_q / q\}_{t \in \mathcal{T}}$ is at most Δ. This provided the following generalization [24] of [5, Theorem 1]:

LEMMA 1. *Let $\omega > 0$ be an arbitrary absolute constant. For a prime q, define*

$$\ell = \left\lceil \omega \left(\frac{\log q \log\log\log q}{\log\log q} \right)^{1/2} \right\rceil \qquad \text{and} \qquad d = \lceil 3 \log q / \ell \rceil .$$

Let $\mathcal{T}$ be a $2^{-\ell}$-homogeneously distributed modulo q sequence of integer numbers. There exists a probabilistic polynomial-time algorithm $\mathcal{A}$ such that for any fixed integer α in the interval $[0, q-1]$, given as input a prime q, d integers $t_1, \ldots, t_d$ and d rationals

$$u_i = APP_{\ell,q}(\alpha t_i), \qquad i = 1, \ldots, d,$$

its output satisfies for sufficiently large q

$$\Pr\left[\mathcal{A}\left(q, t_1, \ldots, t_d; u_1, \ldots, u_d\right) = \alpha\right] \geq 1 - \frac{1}{q}$$

where the probability is taken over all $t_1, \ldots, t_d$ chosen uniformly and independently at random from the elements of $\mathcal{T}$ and all coin tosses of the algorithm $\mathcal{A}$.

Since the previous result applies lattice reduction, it is interesting to know how it is affected if ideal lattice reduction is available, due to the well-known experimental fact that lattice basis reduction algorithms behave much better than theoretically guaranteed, despite NP-hardness results for most lattice problems (see [25, 26]). Nguyen and Shparlinski [24] have obtained the following:

LEMMA 2. *Let $\eta > 0$ be fixed. For a prime q, define $\ell = 1 + \eta$, and*

$$d = \left\lceil \frac{8}{3} \eta^{-1} \log q \right\rceil .$$

Let $\mathcal{T}$ be a $f(q)$-homogeneously distributed modulo q sequence of integer numbers, where $f(q)$ is any function with $f(q) \to 0$ as $q \to \infty$. There exists a deterministic polynomial-time algorithm $\mathcal{A}$ using a CVP_∞-oracle (in dimension $d+1$) such that for any fixed integer α in the interval $[0, q-1]$, given as input a prime q, d integers $t_1, \ldots, t_d$ and d rationals

$$u_i = APP_{\ell,q}(\alpha t_i), \qquad i = 1, \ldots, d,$$

its output satisfies for sufficiently large q

$$\Pr\left[\mathcal{A}\left(q, t_1, \ldots, t_d; u_1, \ldots, u_d\right) = \alpha\right] \geq 1 - \frac{1}{q}$$

where the probability is taken over all $t_1, \ldots, t_d$ chosen uniformly and independently at random from the elements of $\mathcal{T}$.

It is worth noting that in Lemma 2 the assumption on the distribution of $\mathcal{T}$ is quite weak, which explains why in practice, attacks based on variants of the HNP are likely to work (as illustrated in [12, 23]). In fact, only a non-trivial upper bound on the number of fractions $\lfloor at \rfloor_q / q$, $t \in \mathcal{T}$ in a given interval is really needed (rather than the much stronger property of homogeneous distribution modulo q).

We remark that the choice of parameters in DSA and ECDSA is based on the assumption that any attack should take time of order at least $q^{1/2}$. Thus any attack requiring significantly lesser time could still be a threat. Interestingly, one can obtain a combination of Lemma 1 and Lemma 2 which leads to such an attack. We need the following statement which is essentially Lemma 6 from [24]:

LEMMA 3. *For a prime q, define $\ell = \lfloor \log \log q \rfloor$, and*

$$d = \left\lceil 4 \frac{\log q}{\log \log q} \right\rceil .$$

Let $\mathcal{T}$ be a $2^{-\ell}$-homogeneously distributed modulo q sequence of integer numbers. There exists a probabilistic algorithm $\mathcal{A}$ which runs in time $q^{O(1/\log \log q)}$ and such that for any fixed integer α in the interval $[0, q-1]$, given as input a prime q, d integers $t_1, \ldots, t_d$ and d rationals

$$u_i = APP_{\ell,q}(\alpha t_i), \qquad i = 1, \ldots, d,$$

its output satisfies for sufficiently large q

$$\Pr\left[\mathcal{A}\left(q, t_1, \ldots, t_d; u_1, \ldots, u_d\right) = \alpha\right] \geq 1 - \frac{1}{q}$$

where the probability is taken over all $t_1, \ldots, t_d$ chosen uniformly and independently at random from the elements of $\mathcal{T}$.

3. Distribution of Signatures Modulo q

Here we obtain some results about the uniformity of distribution of $t(k, \mu)$ modulo q which can be of independent interest.

Let $\mathbf{e}_p(z) = \exp(2\pi i z/p)$ and $\mathbf{e}_q(z) = \exp(2\pi i z/q)$.

One of our main tools is the *Weil bound* on exponential sums with rational functions which we present in the following form given by Theorem 2 of [21].

LEMMA 4. *For any polynomials $g(X), f(X) \in \mathbb{F}_q[X]$ such that the rational function $F(X) = f(X)/g(X)$ is not constant on $\mathbb{F}_q$, the bound*

$$\left| \sum_{\lambda \in \mathbb{F}_q}^{*} \mathbf{e}_q\left(F(\lambda)\right) \right| \le \left(\max\{\deg g\,, \deg f\} + u - 2\right) q^{1/2} + \delta$$

holds, where $\sum^*$ means that the summation is taken over all $\lambda \in \mathbb{F}_q$ which are not poles of $F(X)$ and

$$(u, \delta) = \begin{cases} (v, 1), & \text{if } \deg f \le \deg g, \\ (v+1, 0), & \text{if } \deg f > \deg g, \end{cases}$$

and v is the number of distinct zeros of $g(X)$ in the algebraic closure of $\mathbb{F}_p$.

We also need some estimates from [16] of exponential sums with points of elliptic curves. The following statement is a very special partial case of Corollary 1 to Theorem 1 of [16].

LEMMA 5. *The bound*

$$\max_{\gcd(c,p)=1} \left| \sum_{k=0}^{q-1} \mathbf{e}_p\left(cx(kG)\right) \right| \le 4p^{1/2}$$

holds.

For an integer $\vartheta \in [0, q-1]$ let us denote by $R(\vartheta)$ be the number of solutions of the equation

$$r(k) = \vartheta, \qquad k \in [1, q-1].$$

LEMMA 6. *The bound*

$$R(\vartheta) = O(p^{1/2} \log p), \qquad \vartheta \in [0, q-1],$$

holds.

Proof. Let

$$L = \left\lfloor \frac{p - \vartheta - 1}{q} \right\rfloor.$$

We remark that $R(\vartheta)$ is the number of solutions $k \in [1, q-1]$ of the congruence

$$x(kG) \equiv qz + \vartheta \pmod{p}, \qquad k \in [1, q-1],\ z \in [0, L].$$

Using the identity (see Exercise 11.a in Chapter 3 of [31])

$$\sum_{c=0}^{p-1} \mathbf{e}_p(cu) = \begin{cases} 0, & \text{if } u \not\equiv 0 \pmod p; \\ p, & \text{if } u \equiv 0 \pmod p; \end{cases}$$

we obtain

$$\begin{aligned} R(\vartheta) &= \frac{1}{p} \sum_{k=1}^{q-1} \sum_{z=0}^{L} \sum_{c=0}^{p-1} \mathbf{e}_p\left(c\left(x(kG) - qz - \vartheta\right)\right) \\ &= \frac{1}{p} \sum_{c=0}^{p-1} \mathbf{e}_p(-c\vartheta) \sum_{k=1}^{q-1} \mathbf{e}_p(cx(kG)) \sum_{z=0}^{L} \mathbf{e}_p(-cqz). \end{aligned}$$

Separating the term

$$\frac{(q-1)(L+1)}{p} \le \frac{(q-1)(p/q+1)}{p} \le 2$$

corresponding to $c = 0$, we derive

$$\begin{aligned} R(\vartheta) &\le 2 + \frac{1}{p} \sum_{c=1}^{p-1} \left| \sum_{k=1}^{q-1} \mathbf{e}_p(cx(kG)) \right| \left| \sum_{z=0}^{L} \mathbf{e}_p(-cqz) \right| \\ &\le 2 + \frac{1}{p} \sum_{c=1}^{p-1} \left| \sum_{k=1}^{q-1} \mathbf{e}_p(cx(kG)) \right| \left| \sum_{z=0}^{L} \mathbf{e}_p(cqz) \right|. \end{aligned}$$

Combining Lemma 5 to estimate the sum over $k \in [1, q-1]$ (certainly the missing term corresponding to $k = 0$ does not change the order of magnitude of this sum) with the estimate

$$\sum_{c=1}^{p-1} \left| \sum_{z=0}^{L} \mathbf{e}_p(cqz) \right| = \sum_{c=1}^{p-1} \left| \sum_{z=0}^{L} \mathbf{e}_p(cz) \right| = O(p \log p),$$

see Exercise 11.c in Chapter 3 of [31], we obtain the desired result. □

In particular, denote by $\mathcal{S}$ the set of pairs $(k, \mu) \in [1, q-1] \times \mathcal{M}$ with $s(k, \mu) \neq 0$ (that is, the set of pairs (k, μ) for which the congruence (1) holds and thus $t(k, \mu)$ is defined). Then

$$|\mathcal{S}| = |\mathcal{M}| \left(q + O\left(p^{1/2} \log p\right)\right). \tag{3}$$

For a hash function $h : \mathcal{M} \to \mathbb{F}_q$ we also denote by W the number of pairs $(\mu_1, \mu_2) \in \mathcal{M}^2$ with $h(\mu_1) = h(\mu_2)$. Thus, $W/|\mathcal{M}|^2$ is a probability of a *collision* and our results are nontrivial under a reasonable assumption that this probability is of order of magnitude close to $1/q$.

First of all, we need to estimate exponential sums with the multipliers $t(k, \mu)$

LEMMA 7. *The bound*

$$\max_{\gcd(c,q)=1} \left| \sum_{(k,\mu)\in\mathcal{S}} \mathbf{e}_q\left(ct(k,\mu)\right) \right| = O\left(W^{1/2}\left(p^{1/2}q^{1/2}\log p + q^{5/4}\right)\right)$$

holds.

Proof. For each $\mu \in \mathcal{M}$ we denote by $\mathcal{K}_\mu$ the set of $k \in [1, q-1]$ for which $(k,\mu) \in \mathcal{S}$.

We consider an element $c_0 \in \mathbb{F}_q^*$ corresponding to the largest exponential sum of our interest. We denote

$$\sigma = \left| \sum_{(k,\mu)\in\mathcal{S}} \mathbf{e}_q\left(c_0 t(k,\mu)\right) \right| = \max_{\gcd(c,q)=1} \left| \sum_{(k,\mu)\in\mathcal{S}} \mathbf{e}_q\left(ct(k,\mu)\right) \right|.$$

We have

$$\sigma \le \sum_{\mu\in\mathcal{M}} \left| \sum_{k\in\mathcal{K}_\mu} \mathbf{e}_q\left(t(k,\mu)\right) \right|.$$

For $\lambda \in \mathbb{F}_q$ we denote by $H(\lambda)$ the number of $\mu \in \mathcal{M}$ with $h(\mu) = \lambda$. We also define the integer $b \in [1, q-1]$ by the congruence $a \equiv 2^{-\ell} c_0 \pmod q$. Then

$$\sigma = \sum_{\lambda\in\mathbb{F}_q} H(\lambda) \left| \sum_{\substack{k=1 \\ \alpha r(k) \not\equiv -\lambda \pmod q}}^{q-1} \mathbf{e}_q\left(a\frac{kr(k)}{\lambda + \alpha r(k)}\right) \right|.$$

Applying the Cauchy inequality we obtain

$$\sigma^2 \le \sum_{\lambda\in\mathbb{F}_q} H(\lambda)^2 \sum_{\lambda\in\mathbb{F}_q} \left| \sum_{\substack{k=1 \\ \alpha r(k) \not\equiv -\lambda \pmod q}}^{q-1} \mathbf{e}_q\left(a\frac{kr(k)}{\lambda + \alpha r(k)}\right) \right|^2. \quad (4)$$

First of all we remark that

$$\sum_{\lambda\in\mathbb{F}_q} H(\lambda)^2 = W. \quad (5)$$

Furthermore,

$$\sum_{\lambda\in\mathbb{F}_q} \left| \sum_{\substack{k=1 \\ \alpha r(k) \not\equiv -\lambda \pmod q}}^{q-1} \mathbf{e}_q\left(a\frac{r(k)}{\lambda + \alpha r(k)}\right) \right|^2$$

$$= \sum_{\lambda \in \mathbb{F}_q} \sum_{\substack{k=1 \\ \alpha r(k) \not\equiv -\lambda \pmod q}}^{q-1}$$
$$\times \sum_{\substack{m=1 \\ \alpha r(m) \not\equiv -\lambda \pmod q}}^{q-1} \mathbf{e}_q\left(a\left(\frac{kr(k)}{\lambda + \alpha r(k)} - \frac{mr(m)}{\lambda + \alpha r(m)}\right)\right)$$
$$= \sum_{k,m=1}^{q-1} {}^*\sum_{\lambda \in \mathbb{F}_q} \mathbf{e}_q\left(a\left(\frac{kr(k)}{\lambda + \alpha r(k)} - \frac{mr(m)}{\lambda + \alpha r(m)}\right)\right),$$

where, as in Lemma 4, the symbol $\sum^*$ means that the summation in the inner sum is taken over all $\lambda \in \mathbb{F}_q$ with

$$\lambda \not\equiv -\alpha r(k) \pmod q \qquad \text{and} \qquad \lambda \not\equiv -\alpha r(m) \pmod q.$$

It is easy to see that if $r(k) \neq r(m)$ then the rational function

$$F_{k,m}(X) = \frac{kr(k)}{X + \alpha r(k)} - \frac{mr(m)}{X + \alpha r(m)}$$

is not constant in $\mathbb{F}_q$. If $r(k) = r(m)$ then

$$F_{k,m}(X) = \frac{(k-m)r(k)}{X + \alpha r(k)}.$$

Thus it is constant only if $k = m$ or $r(k) = r(m) = 0$. From Lemma 6 we see that the number of such pairs is $O\left(p \log^2 p\right)$ for which we estimate the sum over λ trivially as q. For other pairs $(k, m) \in [1, q-1]^2$ we use Lemma 4 getting

$$\sum_{\lambda \in \mathbb{F}_q} \left| \sum_{\substack{k=1 \\ \alpha r(k) \not\equiv -\lambda \pmod q}}^{q-1} \mathbf{e}_q\left(a \frac{r(k)}{\lambda + \alpha r(k)}\right) \right|^2 = O\left(pq \log^2 p + q^{5/2}\right).$$

Substituting this estimate and the identity (5) in (4) we obtain the desired statement. □

LEMMA 8. *The sequence $t(k, \mu)$, $(k, \mu) \in \mathcal{S}$, is Δ-homogeneously distributed modulo q, where*

$$\Delta = O\left(\frac{W^{1/2}\left(p^{1/2} q^{-1/2} \log p + q^{1/4}\right) \log q}{|\mathcal{M}|}\right).$$

Proof. According to a general discrepancy bound, given by Corollary 3.11 of [27] for the discrepancy D of the set

$$\{t(k,\mu) \ : \ (k,\mu) \in \mathcal{S}\}.$$

we have

$$D \le \frac{1}{|\mathcal{S}|} \max_{\gcd(c,q)=1} \left| \sum_{(k,\mu)\in\mathcal{S}} \mathbf{e}_q\left(ct(k,\mu)\right) \right| \log q.$$

and the desired result follows from Lemma 7. □

4. Insecurity of the Elliptic Curve Digital Signature Algorithm

We are ready to prove the main results.

For an integer k we define the oracle $\mathcal{O}_\ell$ such that for any given signature $(r(k), s(k,\mu))$, $k \in [0, q-1]$, $\mu \in \mathcal{M}$, returns the ℓ least significant bits of k.

Combining (2), Lemma 1 and Lemma 8, we obtain

THEOREM 9. *Let $\omega > 0$ be an arbitrary absolute constant. For any $\varepsilon > 0$ there exists $\delta > 0$ such that for any point $G \in \mathbb{E}(\mathbb{F}_p)$ of multiplicative order q, where $q \ge p^{1/2+\varepsilon}$ is prime, and any hash function h with*

$$W \le \frac{|\mathcal{M}|^2}{q^{1-\delta}},$$

given an oracle $\mathcal{O}_\ell$ with

$$\ell = \left\lceil \omega \left(\frac{\log q \log\log\log q}{\log\log q} \right)^{1/2} \right\rceil,$$

there exists a probabilistic polynomial time algorithm to compute the secret parameter α, from $O\left((\log q \log\log q/\log\log\log q)^{1/2}\right)$ signatures $(r(k), s(k,\mu))$ with $k \in [0, q-1]$ and $\mu \in \mathcal{M}$ selected independently and uniformly at random.

Proof. We choose $k \in [0, q-1]$ and $\mu \in \mathcal{M}$ independently and uniformly at random and ignore pairs $(k,\mu) \notin \mathcal{S}$. It follows from (3) that the expected number of choices in order to get d pairs $(k,\mu) \in \mathcal{S}$ is $d + O\left(q^{-\delta}\right)$ for some $\delta > 0$ depending only on $\varepsilon > 0$.

From Lemma 8 we see that the sequence $t(k,\mu)$, $(k,\mu) \in \mathcal{S}$, is Δ-homogeneously distributed with

$$\Delta = O\left(\left(p^{1/2} q^{-1+\delta} \log p + q^{-1/4+\delta}\right) \log q\right) = O(q^{-\delta})$$

for sufficiently small $\delta = \varepsilon/3$.

Now, combining the inequality (2) and Lemma 1 we obtain the result. □

Similarly from (2), Lemma 2 and Lemma 8, we obtain

THEOREM 10. *For any $\varepsilon > 0$ there exists $\delta > 0$ such that for any point $G \in \mathbb{E}(\mathbb{F}_p)$ of multiplicative order q, where $q \geq p^{1/2+\varepsilon}$ is prime, and any hash function h with*

$$W \leq \frac{|\mathcal{M}|^2}{q^{1-\delta}},$$

given an oracle $\mathcal{O}_\ell$ with $\ell = 2$ and a CVP_∞-oracle for the dimension $d+1$ where

$$d = \left\lceil \frac{8}{3} \log q \right\rceil,$$

there exists a probabilistic polynomial time algorithm to compute the secret parameter α, from d signatures $(r(k), s(k,\mu))$ with $k \in [0, q-1]$ and $\mu \in \mathcal{M}$ selected independently and uniformly at random.

Accordingly, from Lemma 3 we derive

THEOREM 11. *For any $\varepsilon > 0$ there exists $\delta > 0$ such that for any point $G \in \mathbb{E}(\mathbb{F}_p)$ of multiplicative order q, where $q \geq p^{1/2+\varepsilon}$ is prime, and any hash function h with*

$$W \leq \frac{|\mathcal{M}|^2}{q^{1-\delta}},$$

given an oracle $\mathcal{O}_\ell$ with

$$\ell = \lceil \log \log q \rceil,$$

there exists a probabilistic algorithm to compute the secret parameter α, in time $q^{O(1/\log\log q)}$, from $O(\log q/\log\log q)$ signatures $(r(k), s(k,\mu))$ with $k \in [0, q-1]$ and $\mu \in \mathcal{M}$ selected independently and uniformly at random.

Using the inequality (2) one can also obtain a similar result for the oracle returning the ℓ most significant bits of k. Oracles returning ℓ consecutive bits in the middle can be studied as well, see the discussion below and in [24].

5. Remarks

First of all we note that the condition $W \leq |\mathcal{M}|^2 q^{-1+\delta}$ does not seem too restrictive. In fact one could expect $W \sim |\mathcal{M}|^2 q^{-1}$ from any "good" hash function.

Our results show that, as in the case of DSA, see [24] it is crucial for the security of ECDSA that the nonce k be generated by a cryptographically secure and unbiased pseudo-random number generator. Even in this case, if the attacker is able to apply a *timing* or *power* attack and select signatures corresponding to small values of k then the whole signature scheme is insecure. Generally, any leakage of information on k could prove dramatic.

It could be useful to remark that Lemma 6 implies that $r(k)$ takes exponentially many distinct values. Thus ECDSA indeed generates exponentially many distinct signatures. Certainly this fact has never been doubted in practice but our results provide its rigorous confirmation.

The same technique can also be applied to designing provable attacks on other modifications of DSA, including the Nyberg–Rueppel signature scheme [9].

Finally, as shown in [24], our results can be generalized to the case of consecutive bits at a known position. The simplest case is when the consecutive bits are the most significant bits. The definition of most significant bits may depend on the context, as opposed to least significant bits. Two possible definitions have been studied in [24]. The usual definition refers to the binary encoding of elements in $\mathbb{F}_q$, where each element is encoded with n bits where $n = 1 + \lfloor \log q \rfloor$ is the bit-length of q. Thus, we define the ℓ most significant bits of an element $x \in \mathbb{F}_q$ as the unique positive integer $\mathrm{MSB}_{\ell,q}(x) \in \{0, \ldots, 2^{\ell-1}\}$ such that:

$$x - 2^{n-\ell}\mathrm{MSB}_{\ell,q}(x) \in \{0, \ldots, 2^{n-\ell} - 1\},$$

For instance, the most significant bit is 1 if $x \geq 2^{n-1}$, and 0 otherwise. However, this definition is not very well-suited to modular residues, since the most significant bit $\mathrm{MSB}_{1,q}(x)$ may in fact leak less than one bit of information: if q is very close to 2^{n-1}, then $\mathrm{MSB}_{1,q}(x)$ is most of the time equal to 0. Hence, Boneh and Venkatesan used in [5] another definition of most significant bits, which we will refer to as most significant modular bits. The ℓ most significant modular bits of an element $x \in \mathbb{F}_q$ are defined as the unique integer $\mathrm{MSMB}_{\ell,q}(x)$ such that

$$0 \leq x - \mathrm{MSMB}_{\ell,q}(x) q/2^{\ell} < q/2^{\ell}.$$

For example, the most significant modular bit is 0 if $x < q/2$, and 1 otherwise. The argument of [24] shows that Theorems 9 and 10 also

hold for most significant usual and modular bits, provided that we add one more bit in the case of most significant (usual) bits. For oracles returning ℓ consecutive bits in the middle, one requires twice as many bits (see [24]).

References

1. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd ACM Symp. on Theory of Computation (STOC'2001), Crete*, pages 601–610, 2001.
2. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
3. M. Bellare, S. Goldwasser, and D. Micciancio. "Pseudo-random" number generation within cryptographic algorithms: The DSS case. In *Proc. of Crypto '97*, volume 1294 of *LNCS*. IACR, Springer-Verlag, 1997.
4. D. Bleichenbacher. On the generation of DSS one-time keys. Manuscript. The result was presented at the Monteverita workshop in March, 2001., February 2001.
5. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *Proc. of Crypto '96*, volume 1109 of *LNCS*. IACR, Springer-Verlag, 1996.
6. D. Boneh and R. Venkatesan. Rounding in lattices and its cryptographic applications. In *Proc. of the 8th Symposium on Discrete Algorithms*, pages 675–681. ACM, 1997.
7. D. R. L. Brown. The exact security of ECDSA. Technical report, Dept. of Combinatorics and Optimization, Univ. of Waterloo, 2000. CORR 2000–54.
8. M. Drmota and R. Tichy. *Sequences, discrepancies and applications*. Springer-Verlag, Berlin, 1997.
9. E. El Mahassni, P. Q. Nguyen, and I. E. Shparlinski. The insecurity of Nyberg–Rueppel and other DSA-like signature schemes with partially known nonce. In *Proc. Workshop on Cryptography and Lattices (CALC '01)*, volume 2146 of *LNCS*, pages 97–109. Springer-Verlag, 2001.
10. M. I. González Vasco and I. E. Shparlinski. On the security of Diffie-Hellman bits. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Computational Number Theory (CCNT'99), Singapore*, pages 257–268. Birkhäuser, 2001.
11. M. I. González Vasco and I. E. Shparlinski. Security of the most significant bits of the Shamir message passing scheme. *Math. Comp.*, 71:333–342, 2002.
12. N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. *Design, Codes and Cryptography*, 23:283–290, 2001.
13. D. Johnson, A. J. Menezes, and S. A. Vanstone. The elliptic curve digital signature algorithm (ECDSA). *Intern. J. of Information Security*, 1:36–63, 2001.
14. N. Koblitz. An elliptic curve implementation of the finite field digital signature algorithm. In *Proc. of Crypto '98*, volume 1462 of *LNCS*, pages 327–337. IACR, Springer-Verlag, 1998.
15. N. Koblitz, A. J. Menezes, and S. A. Vanstone. The state of elliptic curve cryptography. *Designs, Codes and Cryptography*, 19:173–193, 1994.

16. D. Kohel and I. E. Shparlinski. Exponential sums and group generators for elliptic curves over finite fields. In *Algorithmic Number Theory – Proc. of ANTS-IV*, volume 1838 of *LNCS*. Springer-Verlag, 2000.
17. S. V. Konyagin and I. E. Shparlinski. *Character sums with exponential functions and their applications*. Cambridge Univ. Press, Cambridge, 1999.
18. R. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. Wiley-Interscience, NY, 1974.
19. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
20. A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
21. C. J. Moreno and O. Moreno. Exponential sums and Goppa codes, I. *Proc. Amer. Math. Soc.*, 111:523–531, 1991.
22. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 1994.
23. P. Q. Nguyen. The dark side of the hidden number problem: Lattice attacks on DSA. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Computational Number Theory (CCNT'99), Singapore*, pages 321–330. Birkhäuser, 2001.
24. P. Q. Nguyen and I. E. Shparlinski. The insecurity of the Digital Signature Algorithm with partially known nonces. *J. Cryptology*, to appear.
25. P. Q. Nguyen and J. Stern. Lattice reduction in cryptology: An update. In *Algorithmic Number Theory – Proc. of ANTS-IV*, volume 1838 of *LNCS*, pages 85–112. Springer-Verlag, 2000.
26. P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proc. Workshop on Cryptography and Lattices (CALC '01)*, volume 2146 of *LNCS*, pages 146–180. Springer-Verlag, 2001.
27. H. Niederreiter. Quasi-Monte Carlo Methods and Pseudo-random Numbers. *Bull. Amer. Math. Soc.*, 84:957–1041, 1978.
28. H. Niederreiter. *Random Number Generation and Quasi–Monte Carlo Methods*, volume 63. SIAM, Philadelphia, 1992. CBMS-NSF Regional Conference Series in Applied Mathematics.
29. J. H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, 1995.
30. D. R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 1995.
31. I. M. Vinogradov. *Elements of Number Theory*. Dover Publ., New York, 1954.