New Lattice Attacks on DSA Schemes

Dimitrios Poulakis
Department of Mathematics,
Aristotle University of Thessaloniki,
Thessaloniki 54124, Greece,
email:poulakis@math.auth.gr

January 23, 2016

Abstract

We prove that a system of linear congruences of a particular form has at most a unique solution below a certain bound which can be computed efficiently. Using this result we develop attacks against the DSA schemes which, under some assumptions, can provide the secret key in the case where one or several signed messages are available.

MSC 2010: 94A60, 11T71, 11Y16.
Keywords: Public Key Cryptography; Digital Signature Algorithm; Elliptic Curve Digital Signature Algorithm; Closest Vector Problem; Discrete Logarithm.

1 Introduction

In August 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed an algorithm for digital signatures. The algorithm is known as DSA, for Digital Signature Algorithm [18, 16, 14]. It is an efficient variant of the ElGamal digital signature scheme [6] intended for use in electronic mail, electronic funds transfer, electronic data interchange, software distribution, data storage, and other applications which require data integrity assurance and data authentication. In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized [9, 13, 14]. In the next subsection we shall recall the outlines of DSA and ECDSA.

1.1 The DSA Schemes

First, for DSA, the signer chooses a prime p of size between 512 and 1024 bits with increments of 64, q is a prime of size 160 with $q|p-1$ and g is a generator of the unique order q subgroup G of $\mathbb{Z}_p^*$. Further, he chooses

$a \in \{1, \ldots, q-1\}$ and computes $A = g^a \bmod p$. The public key of the signer is (p, q, g, A) and his private key a. Furthermore, the signer chooses a publicly known hash function h mapping messages to $\{0, \ldots, q-1\}$. To sign a message m, he chooses a random number $k \in \{1, \ldots, q-1\}$ which is the ephemeral key, computes

$$r = (g^k \bmod p) \bmod q \quad \text{and} \quad s = k^{-1}(h(m) + ar) \bmod q.$$

The signature of m is the pair (r, s). The signature is valid if and only if the following equality holds:

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

The ECDSA uses an elliptic curve E over $\mathbb{F}_p$ and a point $P \in E(\mathbb{F}_p)$ with order a prime q of size around 160 bits. The signer selects $a \in \{1, \ldots, q-1\}$ and computes $Q = aP$. Its public key is (p, E, P, q, Q) and his private key a. To sign a message m having hash value $h(m) \in \{0, \ldots, q-1\}$, he selects a random number $k \in \{1, \ldots, q-1\}$ which is the ephemeral key and computes $kP = (x, y)$ (where x and y are regarded as integer between 0 and $p-1$). Next, he computes

$$r = x \bmod q \quad \text{and} \quad s = k^{-1}(h(m) + ar) \bmod q.$$

The signature of m is the pair (r, s). For the verification of the signature one computes

$$u_1 = s^{-1}h(m) \bmod q, \quad u_2 = s^{-1}r \bmod q, \quad u_1P + u_2Q = (x_0, y_0).$$

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

The security of DSA (respectively ECDSA) relies on the hardness of the discrete logarithm problem in prime fields and theirs subgroups (respectively in the group of elliptic curve). Thus, the parameters of the two systems were chosen in such a way that the computation of discrete logarithms is computationally infeasible, and so the secret key a and the ephemeral key k are well protected.

1.2 Related Works

The use of lattices and the LLL reduction method [15] is an efficient tool for attacking a variety of cryptographic schemes. Attacks to DSA schemes using lattice reduction techniques and the equality $s = k^{-1}(h(m) + ar) \bmod q$ are given in [1], [12], [19], [20], [3], [21], [5] and [7].

In [1] it is shown that if random numbers for DSA are generated using a linear congruential pseudorandom number generator (LCG), then the combination of the DSA "signature equations" with the LCG generation equations lead to a system of equations which provide the secret key. Babai's closest

vector approximation algorithm is used to solve such a system. This algorithm only returns approximations to the closest vector which are not very good.

In [12], several heuristic attacks to recover the secret key are proposed under the hypothesis that for a reasonable number of signatures, a small fraction of the corresponding nonce k is revealed. As in [1], the attacks are based on the LLL-based Babai CVP approximation algorithm. They used several heuristic assumptions which did not allow precise statements on its theoretical behaviour.

The first provable polynomial-time attack against DSA is given in [19] under the hypothesis that the size of q is not too small compared with p and the probability of collisions for the hash function h is not too large compared with $1/q$. More precisely, if for a certain (polynomially bounded) number of random messages m and random nonces k, about $\log^{1/2} q$ least significant bits of k are known, then in polynomial time one can recover the signer's secret key a. This attack is adapted to the case of ECDSA [20].

In [3], the authors, using one message, compute with the LLL reduction method, two short vectors of a three-dimensional lattice and in case where the second shortest vector is sufficiently short, they deduce two lines which intersect in (a, k), provided that a and k are sufficiently small. If two messages are available one has a linear congruence relating the corresponding ephemeral keys and the same attack is applicable.

The attack presented in [21] uses also one message. The algorithm LLL and two algorithms for the computation of the integral points of two classes of conics are combined for the computation of the secret key provided that at least the elements of one of the sets $\{a, k^{-1} \bmod q\}$, $\{k, a^{-1} \bmod q\}$ and $\{a^{-1} \bmod q, k^{-1} \bmod q\}$ are sufficiently small. As in the previous attack, if two messages are available we can apply these attacks to the congruence relating the two ephemeral keys.

In [5] a two dimensional lattice L is used which is defined by a signed message. Lagrange Lattice Reduction algorithm, provides a basis of L formed by two successive minima. Using this basis we construct two linear polynomials $f_i(x, y)$ such that (a, k) is the intersection point of two straight lines of the form $f_i(x, y) = c_i q$, where $c_i \in \mathbb{Z}$ $(i = 1, 2)$. If a and k are sufficiently small, then c_i belong to a small set and so we can compute the secret key a in polynomial time. Similar attacks hold for the pairs $(k^{-1} \bmod q, k^{-1}a \bmod q)$ and $(a^{-1} \bmod q, a^{-1}k \bmod q)$. If we have two signed messages, then we can apply the same attacks to the equation related the two ephemeral keys.

Finally, in [7] it is assumed that we only know that there are equalities between δ bits of the unknown ephemeral keys used to sign some messages. It is shown that this implicit information should be extracted by constructing a lattice which contains a very short vector such that its components yield the secret key. When the ephemeral keys share enough bits δ, this vector is small enough and so can be computed by the LLL lattice reduction

algorithm. Furthermore, the Gaussian heuristic is used to find a condition on the number of shared bits δ in function of the number of messages for this vector to be the shortest of the lattice.

1.3 Our Contribution

In this paper, we also use the equality $s = k^{-1}(h(m) + ar) \bmod q$ and we develop new attacks on the DSA schemes. More precisely, we first compute a lower bound for the size of vectors of a particular class of lattices. Using this result and the Micciancio-Voulgaris theorem [17] on closest vector problem, we prove that a system of n linear congruences, where $n \leq \log\log q - 1$, of a particular form has at most a unique solution of size smaller than $q^{n/(n+1)}/16$ which can be computed in polynomial time. Thus, using only a signed message, we can construct a such system of linear congruences, which, in the case where it satisfies some condition, provide us the secret key in polynomial time. If we have two signed messages, then we can perform the attack to the linear congruence relating the corresponding ephemeral keys. Further, we give some variants of this attacks using the modular inverses of the keys.

When more than one messages are available it is more possible to construct the linear system with the required properties. If we possess a small fixed number of signed messages, say ≤ 20, the time complexity of the attack remain polynomial. On the other hand, if we use the maximum number of $\log\log q - 1$ messages, then our attack has subexponential running time.

Our attack is rigorous while the attacks in [12] and [7] use heuristic assumptions. Furthermore, it is independent from the pseudorandom number generator which is used for the generation of the random numbers. The attacks in [19, 20] use polynomially bounded number of random messages and in [7] some messages which are signed with ephemeral keys sharing enough bits, while our attacks can be performed with only one message. The attacks in [21] and [5] use also one message but they compute the secret key in polynomial time in the case where at least one of the keys or its modular inverse is at most $q^{1/2} + O(\log q)$. This bound in our attack is $q^{n/(n+1)}/16$. On the other hand, a disadvantage of our attacks is that some hypothesis which are dependent from the available signed messages must be verified that in many cases are not. Finally, note that when ours attacks are applicable we can compute a secret key of full size in the case where its inverse is less than $q^{n/(n+1)}/4$ as the example at the end of the paper shows. Thus, in order the DSA schemes to be protected from ours attacks the secret, the ephemeral keys and their modular inverses must have the same size as q.

1.4 The Structure of the Paper

The paper is organized as follows. In Section 2, we recall some basic results about lattices, the Micciancio-Voulgaris theorem on closest vector problem and we prove a key lemma for the proof of our result. In section 3, using these results, we prove our theorem on the solutions of a system of linear congruences. Our attacks based on it are presented in Sections 4. An example is given in Section 5 and finally Section 6 concludes the paper.

2 Auxiliary Results

Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ be a basis of $\mathbb{R}^n$. A *n-dimensional lattice* spanned by B is the set

$$\mathcal{L} = \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n / \ z_1, \ldots, z_n \in \mathbb{Z}\}.$$

The *Euclidean norm* of a vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$ is defined to be the quantity

$$\|\mathbf{v}\| = (v_1^2 + \cdots + v_n^2)^{1/2}.$$

Throughout the paper we state log for the natural logarithm and $\log_2$ for the logarithm with base 2.

We shall use the following results.

Lemma 1 *Let q be a prime > 2 and integers n with $0 < n \leq \log\log q - 1$, A_i $(i = 1, \ldots, n)$ with $2^{i-1}q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$ and $B_i \in \{1, \ldots, q-1\}$. We denote by $\mathcal{L}$ the lattice spanned by the rows of the square matrix*

$$J = \begin{pmatrix} -1 & A_1 & A_2 & \ldots & A_n \\ 0 & q & 0 & \ldots & 0 \\ 0 & 0 & q & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & q \end{pmatrix}.$$

Then for every nonzero $\mathbf{v} \in \mathcal{L}$ we have

$$\|\mathbf{v}\| > \frac{q^{n/(n+1)}}{8}.$$

Proof. First, we shall see that $A_n < q$. We have $2^n q^{n/(n+1)} < q$ if and only if $2^n < q^{1/(n+1)}$ which is equivalent to $n(n+1) < \log_2 q$. Further,

$$n(n+1) \leq (\log\log q - 1)(\log\log q) \leq (\sqrt{\log q} - 1)\sqrt{\log q} < \log_2 q.$$

It follows that $A_n < q$.

Suppose next that there are integers $x_0, \ldots, x_n$ such that

$$\mathbf{v} = (-x_0, x_0A_1 + x_1q, \ldots, x_0A_n + x_nq)$$

is a nonzero vector of $\mathcal{L}$ satisfying

$$\|\mathbf{v}\| \leq \frac{q^{n/(n+1)}}{8}.$$

Then we have

$$\max\{|x_0|, |x_0A_1 + x_1q|, \ldots, |x_0A_n + x_nq|\} < \frac{q^{n/(n+1)}}{8}.$$

If $x_0 = 0$, then there is $i \in \{1, \ldots, n\}$ such that $x_i \neq 0$ and so, we get $\|\mathbf{v}\| > q$ which is a contradiction. Thus $x_0 \neq 0$. If $x_0A_j + x_jq = 0$, then $x_j \neq 0$. Thus $q|x_0A_j$ and so, we get either $q|x_0$ or $q|A_j$. On the other hand we have $0 < |x_0| < q$ and $0 < A_j < A_n < q$ which is a contradiction. Hence $x_0A_j + x_jq \neq 0$ $(j = 1, \ldots, n)$.

Let $i \in \{1, \ldots, n\}$ such that

$$\frac{q^{(n-i)/(n+1)}}{2^{i+2}} < |x_0| < \frac{q^{(n-i+1)/(n+1)}}{2^{i+1}}.$$

If $x_j \neq 0$ with $1 \leq j \leq i$, then we have

$$\frac{q^{n/(n+1)}}{8} > |x_0A_j + x_jq| \geq q - |x_0|A_j > q - \frac{q^{(n-i+1)/(n+1)}}{2^{i+1}}A_i > \frac{q}{2},$$

whence we get $1 > 4q^{1/(n+1)}$ which is a contradiction. Hence $x_j = 0$ $(j = 1, \ldots, i)$. Then we have

$$\|\mathbf{v}\| > |x_0A_i| > \frac{q^{(n-i)/(n+1)}}{2^{i+2}}2^{i-1}q^{i/(n+1)} = \frac{q^{n/(n+1)}}{8}.$$

which is a contradiction. The result follows.

Theorem 1 *Let $\mathcal{L}$ be a n-dimensional lattice and $\mathbf{y} \in \mathbb{R}^n$. Then there is a deterministic algorithm that computes $\mathbf{v} \in \mathcal{L}$ such that for every $\mathbf{t} \in \mathcal{L}$ we have*

$$\|\mathbf{v} - \mathbf{y}\| \leq \|\mathbf{t} - \mathbf{y}\|$$

in time $2^{2n+o(n)}$.

Proof. See [17].

3 A System of Linear Congruences

Our attacks are based on the following theorem:

Theorem 2 *Let q be a prime > 2 and positive integers n and A_i, B_i $(i = 1, \ldots, n)$ such that $n \leq \log\log q - 1$ and $2^{i-1}q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$. 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 $\mathbf{v} = (x, y_1, \ldots, y_n) \in \{0, \ldots, q-1\}^{n+1}$ having

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

If a such solution exists, then the algorithm of Theorem 1 applied on the vector $(0, B_1, \ldots, B_n)$ and the lattice of Lemma 1 gives a vector $\mathbf{w}$ whose first coordinate is x. The time complexity of computation of x is $O((\log q)^2)$.

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

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

Consider the lattice $\mathcal{L}$ spanned by the rows of the square matrix

$$J = \begin{pmatrix} -1 & A_1 & A_2 & \ldots & A_n \\ 0 & q & 0 & \ldots & 0 \\ 0 & 0 & q & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & q \end{pmatrix}$$

and set $\mathbf{b} = (0, B_1, \ldots, B_n)$. Since $y_i + A_i x + B_i \equiv 0 \,(\mathrm{mod}\, q)$, there is $z_i \in \mathbb{Z}$ such that $-xA_i + z_i q = y_i + B_i$ $(i = 1, \ldots, n)$. Then the vector $\mathbf{u} = (x, -xA_1 + z_1 q, \ldots, -xA_n + z_n q)$ belongs to $\mathcal{L}$ and we have

$$\|\mathbf{u} - \mathbf{b}\| = \|(x, y_1, \ldots, y_n)\| = \|\mathbf{v}\| < \frac{q^{n/(n+1)}}{16}.$$

On the other hand, by Theorem 1 we can compute $\mathbf{w} \in \mathcal{L}$ such that

$$\|\mathbf{w} - \mathbf{b}\| \leq \|\mathbf{u} - \mathbf{b}\| < \frac{q^{n/(n+1)}}{16}.$$

The time complexity of the computation is $2^{2n+o(n)} = O((\log q)^2)$. Thus, we have

$$\|\mathbf{u} - \mathbf{w}\| \leq \|\mathbf{u} - \mathbf{b}\| + \|\mathbf{w} - \mathbf{b}\| < \frac{q^{n/(n+1)}}{8}.$$

Since $\mathbf{u} - \mathbf{w} \in \mathcal{L}$, Lemma 1 implies $\mathbf{u} = \mathbf{w}$. Thus, if $\mathbf{w} = (w_0, \ldots, w_n)$, then we get $x = w_0$.

4 Our Attacks

In this section we describe our attacks. Let n be a positive integer $\leq \log\log q - 1$. Suppose we have $t \leq n$ signed messages m_j $(j = 1, \ldots, t)$ and theirs signatures (r_j, s_j), respectively, with DSA (resp. ECDSA). Then there are $k_j \in \{1, \ldots, q-1\}$ such that $r_j = (g^{k_j} \bmod p) \bmod q$ (resp. $r_j = x_j \bmod q$ and $k_j P = (x_j, y_j)$) and $s_j = k_j^{-1}(h(m_j) + ar_j) \bmod q$. It follows that

$$k_j + C_j a + D_j \equiv 0 \pmod{q} \quad (j = 1, \ldots, t)$$

where $C_j = -r_j s_j^{-1} \bmod q$ and $D_j = -s_j^{-1} h(m_j) \bmod q$.

We give below an algorithm based on Theorem 2 which computes a under some assumptions.

DSA-ATTACK-1
Input: (m_j, r_j, s_j) $(j = 1, \ldots, t)$.

1. Compute $C_j = -r_j s_j^{-1} \bmod q$ and $D_j = -s_j^{-1} h(m_j) \bmod q$.
2. Select integers A_i $(i = 1, \ldots, n)$ with $2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$. (If $2^{i-1} q^{i/(n+1)} < C_i < 2^i q^{i/(n+1)}$, then we can take $A_i = C_i$).
3. Compute $B_{ij} = A_i D_j C_j^{-1} \bmod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$.
4. Denote by M the set of maps $\mu : \{1, \ldots, n\} \to \{1, \ldots, t\}$. For every $\mu \in M$, compute (as it is explained in Theorem 2) the coefficient x_μ of the solution $\mathbf{v}_\mu = (x_\mu, y_{1,\mu(1)}, \ldots, y_{n,\mu(n)})$ of the system

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

   with $\|\mathbf{v}_\mu\| < q^{n/(n+1)}/16$.
5. For every $\mu \in M$ check if x_μ is the private key a.

The performance of the algorithm is given in the following proposition.

Proposition 1 *Put* $k_{ij} = k_j A_i C_j^{-1} \bmod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. *Suppose there is* $\mu \in M$ *such that*

$$\|(a, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < \frac{q^{n/(n+1)}}{16}.$$

Then the algorithm DSA-ATTACK-1 computes a. *Its time complexity for the case of DSA is* $O((\log p)^2 (\log q)^{1+\log t})$ *bit operations and for ECDSA* $O((\log q)^2((\log q)^{\log t} + (\log\log q)^4))$ *bit operations and* $O((\log q)^{1+\log t})$ *elliptic curves operations.*

Proof. Steps 1, 2, 3 and 4 build for every $\mu \in M$ the system of congruences

$$y_i + A_i x + B_{i,\mu(i)} \equiv 0 \,(\mathrm{mod}\, q) \quad (i = 1, \ldots, t).$$

Step 1 needs $O((\log q)^2 t)$ bit operations. The computation of each q^i needs time $O(i^2(\log q)^2)$. The required time for the computation of $\lfloor q^{i/(n+1)} \rfloor$ is $O(i^2(\log q)^2)$ [4] and for $2^i \lfloor q^{i/(n+1)} \rfloor$ is $O(i^3(\log q)^2)$. Thus, we can select A_i such that

$$2^{i-1} \lfloor q^{i/(n+1)} \rfloor + 2^{i-1} \leq A_i \leq 2^i \lfloor q^{i/(n+1)} \rfloor.$$

So, the time complexity of Step 2 is

$$O(\sum_{i=1}^{n} i^3 (\log q)^2) = O((\log q)^2 n^4) = O((\log q)^2 (\log \log q)^4).$$

Step 3 needs $O((\log q)^2 (\log \log q)^2)$ bit operations. Step 4 requires t^n times the application of algorithm of Theorem 1 and so its time complexity is $O((\log q)^{2+\log t})$. Therefore, the time complexity of the construction of all the systems and the computation of x_μ is

$$O((\log q)^2 ((\log q)^{\log t} + (\log \log q)^4)).$$

The complexity of Step 5, in the case of DSA, is $O((\log p)^2 (\log q)^{1+t})$ bit operations and in the case of ECDSA, is $O((\log q)^{1+t})$ elliptic curves operations. Therefore, for DSA, the time complexity of the procedure is $O((\log p)^2 (\log q)^{1+t})$ bit operations and for ECDSA $O((\log q)^2 ((\log q)^{\log t} + (\log \log q)^4))$ bit operations and $O((\log q)^{1+t})$ elliptic curves operations.

For every $\mu \in M$, the vector $(a, k_{1,\mu(1)}, \ldots, k_{n,\mu(n)})$ is a solution of the system of congruences

$$y_i + A_i x + B_{i,\mu(i)} \equiv 0 \,(\mathrm{mod}\, q) \quad (i = 1, \ldots, t).$$

Thus, in the case where there is $\mu \in M$ satisfying

$$\|(a, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < \frac{q^{n/(n+1)}}{16},$$

Theorem 2 implies that a is among the elements x_μ and so it can be computed by the algorithm DSA-ATTACK-1 .

Multiplying by a^{-1} the congruence

$$k_j + C_j a + D_j \equiv 0 \pmod{q} \quad (j = 1, \ldots, t)$$

we get

$$k_j a^{-1} + C_j + D_j a^{-1} \equiv 0 \pmod{q} \quad (j = 1, \ldots, t).$$

Thus, replacing (C_j, D_j) by (D_j, C_j) and a by a^{-1}, we obtain a variant of DSA-ATTACK-1 which under some assumption provides us $a^{-1} \mod q$ and so a. We call it DSA-ATTACK-2. Thus, we have the following result.

Proposition 2 *Put* $k_{ij} = k_j a^{-1} A_i D_j^{-1} \mod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. *Suppose there is* $\mu \in M$ *such that*

$$\|(a^{-1} \mod q, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < \frac{q^{n/(n+1)}}{16},$$

Then the algorithm DSA-ATTACK-2 computes a. *Its time complexity for the case of DSA is* $O((\log p)^2 (\log q)^{1+\log t})$ *bit operations and for ECDSA* $O((\log q)^2((\log q)^{\log t} + (\log\log q)^4))$ *bit operations and* $O((\log q)^{1+\log t})$ *elliptic curves operations.*

Suppose now that $t \geq 2$. So we can eliminate a among the congruences

$$k_j + C_j a + D_j \equiv 0 \pmod q \quad (j = 1, \ldots, t)$$

and we deduce the congruences

$$k_j + \tilde{C}_j k_t + \tilde{D}_j \equiv 0 \pmod q \quad (j = 1, \ldots, t-1),$$

where $\tilde{C}_j = -C_j C_t^{-1} \mod q$ and $\tilde{D}_j = -C_j C_t^{-1} D_t + D_j \mod q$. Replacing in DSA-ATTACK-1 (C_j, D_j) by $(\tilde{C}_j, \tilde{D}_j)$ we have another variant of DSA-ATTACK-1 which we call DSA-ATTACK-3 and under some assumption provide us a. Hence, we have the following result.

Proposition 3 *Put* $k_{ij} = k_j A_i \tilde{C}_j^{-1} \mod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t-1)$. *Suppose there is* $\mu \in M$ *such that*

$$\|(k_t, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < \frac{q^{n/(n+1)}}{16},$$

Then the algorithm DSA-ATTACK-3 computes a. *Its time complexity for the case of DSA is* $O((\log p)^2 (\log q)^{1+\log t})$ *bit operations and for ECDSA* $O((\log q)^2((\log q)^{\log t} + (\log\log q)^4))$ *bit operations and* $O((\log q)^{1+\log t})$ *elliptic curves operations.*

Furthermore, multiplying by k_t^{-1} the congruences

$$k_j + \tilde{C}_j k_t + \tilde{D}_j \equiv 0 \pmod q \quad (j = 1, \ldots, t-1)$$

we obtain

$$k_j k_t^{-1} + \tilde{C}_j + \tilde{D}_j k_t^{-1} \equiv 0 \pmod q \quad (j = 1, \ldots, t-1).$$

So, replacing (C_j, D_j) in DSA-ATTACK-2 by $(\tilde{C}_j, \tilde{D}_j)$ we have another attack which we call DSA-ATTACK-4.

Proposition 4 *Put* $k_{ij} = k_j k_t^{-1} A_i \tilde{D}_j^{-1} \mod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. *Suppose there is* $\mu \in M$ *such that*

$$\|(k_t^{-1} \mod q, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < \frac{q^{n/(n+1)}}{16},$$

Then the algorithm DSA-ATTACK-4 computes a. Its time complexity for the case of DSA is $O((\log p)^2 (\log q)^{1+\log t})$ *bit operations and for ECDSA* $O((\log q)^2((\log q)^{\log t} + (\log\log q)^4))$ *bit operations and* $O((\log q)^{1+\log t})$ *elliptic curves operations.*

Remark 1 If t is very small fixed number, say $t < 20$, then the time complexity of our attacks is polynomial.

Remark 2 Note that as large the number of available messages is, as possible is to construct a system with the required properties.

5 An Example

We consider the elliptic curve E given in [2, Example 3, p. 182] defined over the finite field $\mathbb{F}_p$, where $p = 2^{160} + 7$ is a prime, by the equation

$$y^2 = x^3 + 10x + 1343632762150092499701637438970764818528075565078.$$

The number of points of $E(\mathbb{F}_p)$ is the 160-bit prime

$$q = 1461501637330902918203683518218126812711137002561.$$

Consider the point $P = (x(P), y(P))$ of $E(\mathbb{F}_p)$ of order q, where

$$x(P) = 858713481053070278779168032920613680360047535271,$$

$$y(P) = 364938321350392265038182051503279726748224184066.$$

We take as private key the $160-$bit integer

$$a = 87498466803221173331138684130667374933323658617 8.$$

The public key is $Q = aP = (x(Q), y(Q))$ where

$$x(Q) = 597162246892872056034315330452950636324741691536,$$

$$y(Q) = 118187732920835306056696926675892475754968435739 0.$$

Let m_1, m_2 and m_3 be three messages with hash values

$$\begin{aligned} h(m_1) &= 1238458437157734227527825004718505271235024916418, \\ h(m_2) &= 1028653949698644928576637572550961266718086213222, \\ h(m_3) &= 1359253753908721564345086919389145449479510713328. \end{aligned}$$

Suppose that the following ephemeral keys have been used respectively for the generation of the signatures of the three messages:

$$\begin{aligned} k_1 &= 46608054332288968883546711583551839882652375003 1, \\ k_2 &= 73075081866545145910184241635814150982796627158 9, \\ k_3 &= 73075081866545145910184241635814150982796627968 1. \end{aligned}$$

The size of k_1 is 158 bits and the size of k_2 and k_3 is 159 bits. We have the points $R_i = k_i P = (x(R_i), y(R_i))$ $(i = 1, 2, 3)$, where

$$\begin{aligned} x(R_1) &= 12541577290894439954181238325238082770313139494 62, \\ y(R_1) &= 23109942117176529567525517253616649087109941040, \\ x(R_2) &= 72514437791024688553461670675669940419550766323 1, \\ y(R_2) &= 72483417461458816085624048000585537993089771201 3, \\ x(R_3) &= 25059359814785811483691313826556491545746471085 1, \\ y(R_3) &= 63119281333557571230379851501639067328261656282. \end{aligned}$$

The signature of m_i is (r_i, s_i) where $s_i = k_i^{-1}(h(m_i) + ar_i) \bmod q$ and $r_i = x(R_i)$ $(i = 1, 2, 3)$. We have

$$\begin{aligned} s_1 &= 13638053413353563528076508236901545526539144511 19, \\ s_2 &= 12866440683120842244679891934367692654717672845 71, \\ s_3 &= 13572355400517812931437202327527518406772477540 90. \end{aligned}$$

First, we remark that

$$a^{-1} \bmod q = 507060240091291760598681282150 9 < 2^{103}.$$

Thus, we shall apply DSA-ATTACK-2 with $t = n = 3$. The couple $(a^{-1} \bmod q, k_j a^{-1} \bmod q)$ is a solution of the congruence

$$y + D_i x + C_i \equiv 0 \pmod{q} \quad (i = 1, 2, 3),$$

where

$$\begin{aligned} C_1 &= 14615014631063310496113498840181248212123020995 15, \\ D_1 &= 34359738369, \\ C_2 &= 85658522719296956738171497340749915796614911742 2, \\ D_2 &= 13897735657605247813521742970916786389558362744 32, \\ C_3 &= 25289181258142448854230843836548288088082171610, \\ D_3 &= 49439318646661636536906563016959210019286298249 2. \end{aligned}$$

We have

$$\begin{aligned} \lfloor q^{1/4} \rfloor &= 1099511627775, \\ \lfloor q^{1/2} \rfloor &= 12089258196146291747061 75, \\ \lfloor q^{3/4} \rfloor &= 13292279957849158729038061636335131 55. \end{aligned}$$

Next, we have to consider integers A_i $(i = 1, 2, 3)$ satisfying

$$2^{i-1}q^{i/4} < A_i < 2^i q^{i/4} \quad (i = 1, 2, 3).$$

So, we take $A_1 = D_1$, $A_2 = 2^{81} + 1$ and $A_3 = 2^{122} + 23$.

Since we have

$$\begin{aligned} l_1 &= a^{-1}k_1 \bmod q < 2^{91}, \\ l_2 &= k_2 a^{-1} A_2 D_2^{-1} \bmod q < 2^{90}, \\ l_3 &= k_3 a^{-1} A_3 D_3^{-1} \bmod q < 2^{50}, \end{aligned}$$

we obtain

$$\|(a^{-1} \bmod q, l_1, l_2, l_3)\| < \frac{q^{3/4}}{16}.$$

Hence, the DSA-ATTACK-2 can provide us $a^{-1} \bmod q$ and so, the secret key a.

6 Conclusion

In this paper we proposed some attacks on DSA schemes in the case where one or several signed messages are available. Using these messages we built several systems of linear congruences which, in the case where their coefficients and solutions satisfy some inequalities, can provide us the secret key of the scheme. These attacks can also be applied on other signature schemes where the secret and the ephemeral keys are solutions of a linear congruence, such as Schnorr' signature, Heyst-Pedersen signature, GPS, etc [8, 16, 22]. Note that in the case where the secret, the ephemeral keys and their modular inverses have the same size as q, our attacks cannot be applied.

References

[1] M. Bellare, S. Goldwasser and Micciancio, "Pseudo-random" number generation within cryptographic algorithms: the DSS case. In *Proc. of Crypto '97,* LNCS 1294. IACR, Palo Alto, CA. Springer-Verlag, Berlin 1997.

[2] I. F. Blake, G. Seroussi and N. Smart, Elliptic Curves in Cryptography, Cambridge University Press 2000.

[3] I. F. Blake and T. Garefalakis, On the security of the digital signature algorithm, *Des. Codes Cryptogr.,* 26, no. 1-3 (2002), 87-96.

[4] R. Brent and P. Zimmerman, *Modern Computer Arithmetic,* Cambridge University Press 2011.

[5] K. Draziotis and D. Poulakis, Lattice attacks on DSA schemes based on Lagrange's algorithm. 5th international Conference on Algebraic Informatics, CAI 2013. Berlin: Springer. LNCS 8080, 119-131 (2013).

[6] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithm, *IEEE Transactions on Information Theory,* 31 (1985), 469-472.

[7] J.-L. Faugère, C. Goyet, and G. Renault, *Attacking (EC)DSA Given Only an Implicit Hint, Selected Area of Cryptography,* LNCS 7707, pp. 252-274, Springer-Verlag, Berlin - Heidelberg 2013.

[8] M. Girault, G. Poupard and J. Stern, Global Payment System (GPS): un protocole de signature à la volée, *Proceedings of Trusting Electronic Trade,* 7-9 juin 1999.

[9] D. Johnson, A. J. Menezes and S. A. Vastone, The elliptic curve digital signature algorithm (ECDSA), *Intern. J. of Information Security,* 1 (2001) 36-63.

[10] J. Hoffstein, J. Pipher and J. Silverman, An Introduction to Mathematical Cryptography, Springer 2008.

[11] N. A. Howgrave-Graham, *Finding small roots of univariate equations revisited.* In Cryptography and Coding, vol. 1355 of LNCS, pp. 131-142. Springer Verlag, 1997.

[12] N. A. Howgrave-Graham and N. P. Smart, Lattice Attacks on Digital Signature Schemes, *Des. Codes Cryptogr.* 23 (2001) 283-290.

[13] N. Koblitz, A. J. Menezes and S. A. Vastone, The state of elliptic curve cryptography, *Des. Codes Cryptogr.* 19 (2000), 173-193.

[14] N. Koblitz and A. J. Menezes, A survey of Public-Key Cryptosystems, *SIAM REVIEW,* 46, No. 4 (2004), 599-634.

[15] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, Factoring polynomials with rational coefficients, *Math. Ann.*, 261 (1982), 513-534.

[16] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography,* CRC Press, Boca Raton, Florida, 1997.

[17] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. *In Proc. of STOC, ACM*, (2010) pages 351-358.

[18] National Institute of Standards and Technology (NIST). *FIPS Publication* 186*: Digital Signature Standard.* May 1994.

[19] P. Nguyen and I. E. Shparlinski, The Insecurity of the Digital Signature Algorithm with Partially Known Nonces, *J. Cryptology,* 15 (2002), 151-176.

[20] P. Nguyen and I. E. Shparlinski, The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces, *Des. Codes Cryptogr.* 30, (2003), 201-217.

[21] D. Poulakis, Some Lattice Attacks on DSA and ECDSA, *Applicable Algebra in Engineering, Communication and Computing,* 22, (2011), 347-358.

[22] D. R. Stinson, *Cryptography, Theory and Practice,* Chapman & Hall/CRC, 2nd ed. 2002.