A New Lattice Attack on DSA Schemes

Dimitrios Poulakis (Thessaloniki)

June 7, 2014

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

For the construction of a such scheme the signer chooses:

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

For the construction of a such scheme the signer chooses:

- primes p and q such that $q|p-1$, size$(q) = 160, 224, 256$ bits, size$(p) = 1024, 2048, 3072$ bits.

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

For the construction of a such scheme the signer chooses:

- primes p and q such that $q|p-1$, $\text{size}(q) = 160, 224, 256$ bits, $\text{size}(p) = 1024, 2048, 3072$ bits.
- $g \in \{1, \ldots, p-1\}$ with $\mathrm{ord}_p(g) = q$.

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

For the construction of a such scheme the signer chooses:

- primes p and q such that $q|p-1$, size$(q) = 160, 224, 256$ bits, size$(p) = 1024, 2048, 3072$ bits.
- $g \in \{1, \ldots, p-1\}$ with $\mathrm{ord}_p(g) = q$.
- $a \in \{1, \ldots, q-1\}$ and $\mathcal{A} = g^a \bmod p$.

DSA

In 1991, the U.S. government's National Institute of Standards and Technology (NIST) proposed DSA (Digital Signature Algorithm).

For the construction of a such scheme the signer chooses:

- primes p and q such that $q|p-1$, size$(q) = 160, 224, 256$ bits, size$(p) = 1024, 2048, 3072$ bits.
- $g \in \{1, \ldots, p-1\}$ with $\mathrm{ord}_p(g) = q$.
- $a \in \{1, \ldots, q-1\}$ and $\mathcal{A} = g^a \bmod p$.
- an one-way, collision-free hash function $h : \{0,1\}^* \to \{0, \ldots, q-1\}$.

DSA

Parameters : (p, q, g, h)

Public key: $\mathcal{A}$.

Private key: a.

DSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

DSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He selects at random $k \in \{1, \ldots, q-1\}$.

DSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He selects at random $k \in \{1, \ldots, q-1\}$.
- He computes $r = (g^k \bmod p) \bmod q$.

DSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He selects at random $k \in \{1, \ldots, q-1\}$.
- He computes $r = (g^k \bmod p) \bmod q$.
- He computes $s = k^{-1}(h(m) + ar) \bmod q$.

DSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He selects at random $k \in \{1, \ldots, q-1\}$.
- He computes $r = (g^k \bmod p) \bmod q$.
- He computes $s = k^{-1}(h(m) + ar) \bmod q$.

The signature of m is (r, s).

DSA

Verification. The verification of the signed message (m, r, s) is performed by checking

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

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

- an elliptic curve E over $\mathbb{F}_p$,

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

- an elliptic curve E over $\mathbb{F}_p$,
- a prime q with $2^{159} < q < 2^{160}$ and $q \mid |E(\mathbb{F}_p)|$,

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

- an elliptic curve E over $\mathbb{F}_p$,
- a prime q with $2^{159} < q < 2^{160}$ and $q \mid |E(\mathbb{F}_p)|$,
- $P \in E(\mathbb{F}_p)$ with $\mathrm{ord}(P) = q$,

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

- an elliptic curve E over $\mathbb{F}_p$,
- a prime q with $2^{159} < q < 2^{160}$ and $q \mid |E(\mathbb{F}_p)|$,
- $P \in E(\mathbb{F}_p)$ with $\mathrm{ord}(P) = q$,
- $a \in \{1, \ldots, q-1\}$ and computes $Q = aP$,

ECDSA

In 1998, an elliptic curve analogue called Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed and standardized

For the construction of a such scheme the signer chooses

- an elliptic curve E over $\mathbb{F}_p$,
- a prime q with $2^{159} < q < 2^{160}$ and $q \mid |E(\mathbb{F}_p)|$,
- $P \in E(\mathbb{F}_p)$ with $\mathrm{ord}(P) = q$,
- $a \in \{1, \ldots, q-1\}$ and computes $Q = aP$,
- an one-way and collision-free hash function $h : \{0,1\}^* \rightarrow \{0, \ldots, q-1\}$.

ECDSA

Parameters: (p, E, P, q, h)

Public key: Q.

Private key: a.

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He chooses at random $k \in \{1, \ldots, q-1\}$.

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He chooses at random $k \in \{1, \ldots, q-1\}$.
- He computes $kP = (\bar{x}, \bar{y})$ $(x, y \in \{0, \ldots, p-1\})$.

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He chooses at random $k \in \{1, \ldots, q-1\}$.
- He computes $kP = (\bar{x}, \bar{y})$ $(x, y \in \{0, \ldots, p-1\})$.
- He computes $r = x \mod q$.

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He chooses at random $k \in \{1, \ldots, q-1\}$.
- He computes $kP = (\bar{x}, \bar{y})$ $(x, y \in \{0, \ldots, p-1\})$.
- He computes $r = x \mod q$.
- He computes $s = k^{-1}(h(m) + ar) \mod q$.

ECDSA

Signature. To sign a message $m \in \{0,1\}^*$ the signer works as follows:

- He chooses at random $k \in \{1, \ldots, q-1\}$.
- He computes $kP = (\bar{x}, \bar{y})$ $(x, y \in \{0, \ldots, p-1\})$.
- He computes $r = x \bmod q$.
- He computes $s = k^{-1}(h(m) + ar) \bmod q$.

The signature of m is (r, s).

ECDSA

Verification. The verification of the signed message (m, r, s) is performed by computing:

ECDSA

Verification. The verification of the signed message (m, r, s) is performed by computing:

- $u_1 = s^{-1}h(m) \bmod q$,

ECDSA

Verification. The verification of the signed message (m, r, s) is performed by computing:

- $u_1 = s^{-1}h(m) \bmod q$,
- $u_2 = s^{-1}r \bmod q$,

ECDSA

Verification. The verification of the signed message (m, r, s) is performed by computing:

- $u_1 = s^{-1}h(m) \bmod q$,
- $u_2 = s^{-1}r \bmod q$,
- $u_1P + u_2Q = (\bar{x}_0, \bar{y}_0)$.

ECDSA

Verification. The verification of the signed message (m, r, s) is performed by computing:

- $u_1 = s^{-1}h(m) \bmod q$,
- $u_2 = s^{-1}r \bmod q$,
- $u_1P + u_2Q = (\bar{x}_0, \bar{y}_0)$.

The signature is accepted if-if $r = x_0 \bmod q$.

Security

The security of *DSA* is relied on the difficulty of computation of the discrete logarithms a and k from the relations

$$\mathcal{A} = g^a \bmod p$$

and

$$r = (g^k \bmod p) \bmod q.$$

Security

The security of *ECDSA* is relied on the difficulty of computation of the discrete logarithms a and k from the relations

$$Q = aP$$

and

$$kP = (\bar{x}, \bar{y}).$$

Security

Important Remark

In both cases a and k is a solution of the congruence

$$s = k^{-1}(h(m) + ar) \bmod q.$$

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.
4. 2002. I. F. Blake and T. Garefalakis.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.
4. 2002. I. F. Blake and T. Garefalakis.
5. 2003. P. Nguyen and I. E. Shparlinski.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.
4. 2002. I. F. Blake and T. Garefalakis.
5. 2003. P. Nguyen and I. E. Shparlinski.
6. 2011. D. Poulakis.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.
4. 2002. I. F. Blake and T. Garefalakis.
5. 2003. P. Nguyen and I. E. Shparlinski.
6. 2011. D. Poulakis.
7. 2013. J.-L. Faugère, C. Goyet, and G. Renault.

Attacks based on $s = k^{-1}(h(m) + ar) \bmod q$

1. 1997. M. Bellare, S. Goldwasser and Micciancio.
2. 2001. N. A. Howgrave-Graham and N. P. Smart.
3. 2002 P. Nguyen and I. E. Shparlinski.
4. 2002. I. F. Blake and T. Garefalakis.
5. 2003. P. Nguyen and I. E. Shparlinski.
6. 2011. D. Poulakis.
7. 2013. J.-L. Faugère, C. Goyet, and G. Renault.
8. 2013. K. Draziotis and D. Poulakis.

Lattices

Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a basis of $\mathbb{R}^n$.

Lattices

Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a basis of $\mathbb{R}^n$.

A *n-dimensional lattice* spanned by B is the set

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

Lattices

Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ 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 $\mathbf{v} = (v_1, \ldots, v_n)$ is the quantity

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

Closest Vector Problem (CVP)

Problem

Let $\mathcal{L} \subset \mathbb{R}^n$ *be a lattice and* $\mathbf{w} \in \mathbb{R}^n \setminus \mathcal{L}$*. Find a vector* $\mathbf{v} \in \mathcal{L}$ *that minimizes the quantity* $\|\mathbf{v} - \mathbf{w}\|$*.*

CVP is NP-hard problem.

2010. D. Micciancio and P. Voulgaris

Theorem

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

A System of Linear Congruences

Our attacks are based on the following result:

Theorem

Let q be an integer > 0. Consider integers n with $0 < n < \log_2 q$, A_i with

$$2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$$

and $B_i \in \{1, \ldots, q-1\}$. Then the system of congruences

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

has at most one solution $\mathbf{v} = (x, y_1, \ldots, y_n) \in \{0, \ldots, q-1\}^{n+1}$ having

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

The time complexity of computation of x is $O(2^{2n+o(n)})$.

For the proof of this result we use the theorem of Micciancio and P. Voulgaris, and the following lemma:

Lemma

Let q be an integer > 0. Consider integers n and A_i such that $0 < n < \log_2 q$, and $2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$. We denote by $\mathcal{L}$ the lattice spanned by the rows of the square matrix

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

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

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

$$n \leq 2\lfloor \log_2 \log_2 q \rfloor.$$

$n \leq 2\lfloor \log_2 \log_2 q \rfloor$.

m_j messages and (r_j, s_j) theirs signatures with DSA (resp. ECDSA) ($j = 1, \ldots, t \leq n$).

$n \leq 2\lfloor \log_2 \log_2 q \rfloor$.

m_j messages and (r_j, s_j) theirs signatures with DSA (resp. ECDSA) ($j = 1, \ldots, t \leq n$).

$r_j = (g^{k_j} \bmod p) \bmod q$,
(resp. $k_j P = (x_j, y_j)$ and $r_j = x_j \bmod q$).

$n \leq 2\lfloor \log_2 \log_2 q \rfloor$.

m_j messages and (r_j, s_j) theirs signatures with DSA (resp. ECDSA) ($j = 1, \ldots, t \leq n$).

$r_j = (g^{k_j} \bmod p) \bmod q$,
(resp. $k_j P = (x_j, y_j)$ and $r_j = x_j \bmod q$).

$s_j = k_j^{-1}(h(m_j) + a r_j) \bmod q$.

$n \leq 2\lfloor \log_2 \log_2 q \rfloor$.

m_j messages and (r_j, s_j) theirs signatures with DSA (resp. ECDSA) ($j = 1, \ldots, t \leq n$).

$r_j = (g^{k_j} \bmod p) \bmod q$,
(resp. $k_j P = (x_j, y_j)$ and $r_j = x_j \bmod q$).

$s_j = k_j^{-1}(h(m_j) + a r_j) \bmod q$.

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

DSA-ATTACK-1

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

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

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

   $$2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$$

   and denote by $\mathcal{L}$ the lattice spanned by

   $$(-1, A_1, \ldots, A_n),\ (0, q, 0, \ldots, 0), \ldots,\ (0, \ldots, 0, q).$$

   (If $2^{i-1} q^{i/(n+1)} < C_i < 2^i q^{i/(n+1)}$, we can take $A_i = C_i$).

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
$$2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$$
and denote by $\mathcal{L}$ the lattice spanned by
$$(-1, A_1, \ldots, A_n),\ (0, q, 0, \ldots, 0), \ldots,\ (0, \ldots, 0, q).$$
(If $2^{i-1} q^{i/(n+1)} < C_i < 2^i q^{i/(n+1)}$, we can take $A_i = C_i$).
3. Compute $B_{ij} = A_i D_j C_j^{-1} \bmod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. Denote by M the set of maps $\mu : \{1, \ldots, n\} \rightarrow \{1, \ldots, t\}$. For every $\mu \in M$ we set $\mathbf{b}_\mu = (0, B_{1\mu(1)}, \ldots, B_{n\mu(n)})$.

DSA-ATTACK-1

Input: (m_j, r_j, s_j) $(j = 1, \dots, t)$.

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 $(i = 1, \dots, n)$ with
$$2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$$
and denote by $\mathcal{L}$ the lattice spanned by
$$(-1, A_1, \dots, A_n),\ (0, q, 0, \dots, 0), \dots,\ (0, \dots, 0, q).$$
(If $2^{i-1} q^{i/(n+1)} < C_i < 2^i q^{i/(n+1)}$, we can take $A_i = C_i$).
3. Compute $B_{ij} = A_i D_j C_j^{-1} \bmod q$ $(i = 1, \dots, n,\ j = 1, \dots, t)$. Denote by M the set of maps $\mu : \{1, \dots, n\} \to \{1, \dots, t\}$. For every $\mu \in M$ we set $\mathbf{b}_\mu = (0, B_{1\mu(1)}, \dots, B_{n\mu(n)})$.
4. Using the algorithm of Theorem 1, $\forall \mu \in M$ compute $\mathbf{v}_\mu \in \mathcal{L}$ s. t. $\forall \mathbf{t} \in \mathcal{L}$ we have $\|\mathbf{v}_\mu - \mathbf{b}_\mu\| \leq \|\mathbf{t} - \mathbf{b}_\mu\|$.

DSA-ATTACK-1

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

1. Compute $C_j = -r_j s_j^{-1} \mod q$ and $D_j = -s_j^{-1} h(m_j) \mod q$.
2. Select integers A_i $(i = 1, \ldots, n)$ with
$$2^{i-1} q^{i/(n+1)} < A_i < 2^i q^{i/(n+1)}$$
and denote by $\mathcal{L}$ the lattice spanned by
$$(-1, A_1, \ldots, A_n),\ (0, q, 0, \ldots, 0), \ldots,\ (0, \ldots, 0, q).$$
(If $2^{i-1} q^{i/(n+1)} < C_i < 2^i q^{i/(n+1)}$, we can take $A_i = C_i$).
3. Compute $B_{ij} = A_i D_j C_j^{-1} \mod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. Denote by M the set of maps $\mu : \{1, \ldots, n\} \to \{1, \ldots, t\}$. For every $\mu \in M$ we set $\mathbf{b}_\mu = (0, B_{1\mu(1)}, \ldots, B_{n\mu(n)})$.
4. Using the algorithm of Theorem 1, $\forall \mu \in M$ compute $\mathbf{v}_\mu \in \mathcal{L}$ s. t. $\forall \mathbf{t} \in \mathcal{L}$ we have $\|\mathbf{v}_\mu - \mathbf{b}_\mu\| \leq \|\mathbf{t} - \mathbf{b}_\mu\|$.
5. For every $\mu \in M$ check if the first coordinate of $\mathbf{v}_\mu$ is a.

Proposition

Put $k_{ij} = k_j \lfloor q^{i/(n+1)} \rfloor C_j^{-1} \bmod q$ $(i = 1, \ldots, n,\ j = 1, \ldots, t)$. *Then the algorithm DSA-ATTACK-1 computes a provided that*

$$\|(a, k_{1\mu(1)}, \ldots, k_{n\mu(n)})\| < q^{n/(n+1)}/4,$$

where $\mu \in M$. *The time complexity of the algorithm is* $O((\log_2 q)^{4+2\log_2 t})$.

We also have the congruences

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

Replacing (C_j, D_j) by (D_j, C_j) and a by a^{-1}, we obtain a variant of DSA-ATTACK-1 called DSA-ATTACK-2.

We also have the congruences

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

Replacing (C_j, D_j) by (D_j, C_j) and a by a^{-1}, we obtain a variant of DSA-ATTACK-1 called DSA-ATTACK-2.

Suppose $t \geq 2$. We eliminate a among the congruences

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

Setting $\tilde{C}_j = -C_j C_t^{-1} \bmod q$, $\tilde{D}_j = -C_j C_t^{-1} D_j \bmod q$, we get

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

Thus we have another attack called DSA-ATTACK-3.

We also have the congruences

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

Replacing (C_j, D_j) by (D_j, C_j) and a by a^{-1}, we obtain a variant of DSA-ATTACK-1 called DSA-ATTACK-2.

Suppose $t \geq 2$. We eliminate a among the congruences

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

Setting $\tilde{C}_j = -C_j C_t^{-1} \bmod q$, $\tilde{D}_j = -C_j C_t^{-1} D_j \bmod q$, we get

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

Thus we have another attack called DSA-ATTACK-3.

Finally, we have the congruences

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

which give another attack called DSA-ATTACK-4.

An Example

Let E be the elliptic curve defined over $\mathbb{F}_p$, where $p = 2^{160} + 7$ is a prime, by the equation

$$y^2 = x^3 + 10x + C,$$

where

$$C = 1343632762150092499701637438970764818528075565078.$$

An Example

Let E be the elliptic curve defined over $\mathbb{F}_p$, where $p = 2^{160} + 7$ is a prime, by the equation

$$y^2 = x^3 + 10x + C,$$

where

$$C = 1343632762150092499701637438970764818528075565078.$$

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 $E(\mathbb{F}_p)$, where

$$x(P) = 8587134810530702787791680329206136803600475352 71,$$

$$y(P) = 3649383213503922650381820515032797267482241840 66.$$

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

$$x(P) = 858713481053070278779168032920613680360047535271,$$

$$y(P) = 364938321350392265038182051503279726748224184066.$$

We take as private key the 160−bit integer

$$a = 874984668032211733311386841306673749333236586178.$$

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

$$x(P) = 858713481053070278779168032920613680360047535271,$$

$$y(P) = 364938321350392265038182051503279726748224184066.$$

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

$$a = 874984668032211733311386841306673749333236586178.$$

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

$$x(Q) = 597162246892872056034315330452950636324741691536,$$

$$y(Q) = 1181877329208353060566969266758924757549684357390.$$

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

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

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

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

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

$$\begin{aligned} k_1 &= 466080543322889688835467115835518398826523750031, \\ k_2 &= 730750818665451459101842416358141509827966271589, \\ k_3 &= 730750818665451459101842416358141509827966279681. \end{aligned}$$

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

$$
\begin{aligned}
x(R_1) &= 125415772908944399541812383252380827703131394946\,2, \\
y(R_1) &= 23109942117176529567525517253616649087109941040, \\
x(R_2) &= 72514437791024688553461670675669940419550766323\,1, \\
y(R_2) &= 72483417461458816085624048000585537993089771201\,3, \\
x(R_3) &= 25059359814785811483691313826556491545746471085\,1, \\
y(R_3) &= 63119281333557571230379851501639067328261656282.
\end{aligned}
$$

We have the points $R_i = k_i P = (x(R_i), y(R_i))$ $(i = 1, 2, 3)$, where

$$\begin{aligned}
x(R_1) &= 1254157729089443995418123832523808277031313949462, \\
y(R_1) &= 2310994211717652956752551725361664908710994 1040, \\
x(R_2) &= 7251443779102468855346167067566994041955076 63231, \\
y(R_2) &= 7248341746145881608562404800058553799308977 12013, \\
x(R_3) &= 2505935981478581148369131382655649154574647 10851, \\
y(R_3) &= 6311928133355757123037985150163906732826165 6282.
\end{aligned}$$

The signarure 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

$$\begin{aligned}
s_1 &= 1363805341335356352807650823690154552653914451119, \\
s_2 &= 1286644068312084224467989193436769265471767284571, \\
s_3 &= 1357235540051781293143720232752751840677247754090.
\end{aligned}$$

First, we remark that

$$a^{-1} \bmod q = 5070602400912917605986812821509 < 2^{103}.$$

Thus, we shall apply DSA-ATTACK-2 with $n = 3$.

First, we remark that

$$a^{-1} \bmod q = 5070602400912917605986812821509 < 2^{103}.$$

Thus, we shall apply DSA-ATTACK-2 with $n = 3$.

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

First, we remark that

$$a^{-1} \bmod q = 5070602400912917605986812821509 < 2^{103}.$$

Thus, we shall apply DSA-ATTACK-2 with $n = 3$.

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

$$\begin{aligned}
C_1 &= 1461501463106331049611349884018124821212302099515, \\
D_1 &= 34359738369, \\
C_2 &= 856585227192969567381714973407499157966149117422, \\
D_2 &= 1389773565760524781352174297091678638955836274432, \\
C_3 &= 252891812581424488542308438365482880880821716 10, \\
D_3 &= 494393186466616365369065630169592100192862982492..
\end{aligned}$$

We have

$$\begin{aligned} \lfloor q^{1/4} \rfloor &= 1099511627775, \\ \lfloor q^{1/2} \rfloor &= 1208925819614629174706175, \\ \lfloor q^{3/4} \rfloor &= 1329227995784915872903806163633513155. \end{aligned}$$

We have

$$
\begin{aligned}
\lfloor q^{1/4} \rfloor &= 1099511627775, \\
\lfloor q^{1/2} \rfloor &= 1208925819614629174706175, \\
\lfloor q^{3/4} \rfloor &= 1329227995784915872903806163633513155.
\end{aligned}
$$

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

We have

$$\begin{aligned}\lfloor q^{1/4} \rfloor &= 1099511627775, \\ \lfloor q^{1/2} \rfloor &= 1208925819614629174706175, \\ \lfloor q^{3/4} \rfloor &= 1329227995784915872903806163633513155.\end{aligned}$$

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

We have

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

Since we have

$$l_1 = a^{-1}k_1 \mod q < 2^{91},$$
$$l_2 = k_2 a^{-1} A_2 D_2^{-1} \ \mathrm{mod} \ q < 2^{90},$$
$$l_3 = k_3 a^{-1} A_3 D_3^{-1} \ \mathrm{mod} \ q < 2^{50},$$

we obtain

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

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

THANK YOU