

IMPROVED LATTICE ATTACKS ON LEAKY ECDSA

ADAM MADRO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

Objective

- ▶ Our aim is to improve side channel attacks on ECDSA.



Objective

- ▶ Our aim is to improve side channel attacks on ECDSA.
- ▶ I proposed combining two methods for improving lattice attacks. One improves the success rate at the cost of speed, while the other improves the speed.



Objective

- ▶ Our aim is to improve side channel attacks on ECDSA.
- ▶ I proposed combining two methods for improving lattice attacks. One improves the success rate at the cost of speed, while the other improves the speed.
- ▶ I showed formally that these methods interact nontrivially, and have the potential to be faster when combined than the original speed-focused attack, without sacrificing the success rate.



Table of Contents

Problem and Context

My Contribution



UNIVERSITÀ
DEGLI STUDI
DI MILANO



UNIVERSITÄT
DUISBURG
ESSEN
Offen im Denken

- ▶ ECDSA: *Elliptic Curve Digital Signature Algorithm*



Leaky ECDSA

- ▶ ECDSA: *Elliptic Curve Digital Signature Algorithm*
- ▶ To generate a signature, one must generate a random number k called the *nonce*.



Leaky ECDSA

- ▶ ECDSA: *Elliptic Curve Digital Signature Algorithm*
- ▶ To generate a signature, one must generate a random number k called the *nonce*.
- ▶ Attackers can use side channels to obtain a small amount of information about the nonce for each signature.
- ▶ Given many signatures, each with a small amount of nonce leakage, it is known that one can recover the secret key by solving a lattice problem called the *Shortest Vector Problem* (SVP).



Lattices

- ▶ A (discrete/integral) lattice is an additive subgroup of $\mathbb{R}^n$ generated by a finite number of linearly independent vectors.



Lattices

- ▶ A (discrete/integral) lattice is an additive subgroup of $\mathbb{R}^n$ generated by a finite number of linearly independent vectors.
- ▶ The lattice $\mathcal{L}$ generated by a matrix $\mathcal{B} \in \mathbb{Z}^{n \times n}$ (called the *basis* of $\mathcal{L}$) is the set of all linear combinations of the rows of $\mathcal{B}$:
 - ▶ $\mathcal{L} = \{\mathbf{x} \cdot \mathcal{B} \mid \mathbf{x} \in \mathbb{R}^n\}$
- ▶ (Note that vectors are treated as row vectors)



Shortest Vector Problem

- ▶ Given a lattice $\mathcal{L}$, the SVP asks us to find the shortest vector in $\mathcal{L}$.



Shortest Vector Problem

- ▶ Given a lattice $\mathcal{L}$, the SVP asks us to find the shortest vector in $\mathcal{L}$.
- ▶ For our attack on ECDSA:
 - ▶ Using our sampled signatures and leaked data, we construct a lattice which contains a vector whose coordinates encode the secret signing key.



Shortest Vector Problem

- ▶ Given a lattice $\mathcal{L}$, the SVP asks us to find the shortest vector in $\mathcal{L}$.
- ▶ For our attack on ECDSA:
 - ▶ Using our sampled signatures and leaked data, we construct a lattice which contains a vector whose coordinates encode the secret signing key.
 - ▶ Given enough signatures and nonce leakage, we can expect this vector to be the shortest one in the lattice.



Limitations of SVP Attacks

- ▶ The SVP can be solved by algorithms such as *lattice sieving*, but there are limitations.



Limitations of SVP Attacks

- ▶ The SVP can be solved by algorithms such as *lattice sieving*, but there are limitations.
- ▶ Limitation of SVP-Solvers:
 - ▶ Exponentially slow with respect to lattice dimension.



Limitations of SVP Attacks

- ▶ The SVP can be solved by algorithms such as *lattice sieving*, but there are limitations.
- ▶ Limitation of SVP-Solvers:
 - ▶ Exponentially slow with respect to lattice dimension.
- ▶ Limitation of SVP-solving as a method for attacking ECDSA:
 - ▶ To guarantee that the vector containing the secret key is the shortest vector in the lattice, we need many signature samples.
 - ▶ Each signature sample increases the lattice dimension by 1.



Limitations of SVP Attacks

- ▶ The SVP can be solved by algorithms such as *lattice sieving*, but there are limitations.
- ▶ Limitation of SVP-Solvers:
 - ▶ Exponentially slow with respect to lattice dimension.
- ▶ Limitation of SVP-solving as a method for attacking ECDSA:
 - ▶ To guarantee that the vector containing the secret key is the shortest vector in the lattice, we need many signature samples.
 - ▶ Each signature sample increases the lattice dimension by 1.

⇒ Successful attacks take too long!

Table of Contents

Problem and Context

My Contribution



UNIVERSITÀ
DEGLI STUDI
DI MILANO



UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

Dimensions for Free (D4F)

- ▶ Let $\mathcal{L}$ be a lattice with basis $\mathcal{B}$, and let $\pi_{n-d} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-d}$ be the projection of $\mathbb{R}^n$ orthogonal to the first d rows of $\mathcal{B}$.



Dimensions for Free (D4F)

- ▶ Let $\mathcal{L}$ be a lattice with basis $\mathcal{B}$, and let $\pi_{n-d} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-d}$ be the projection of $\mathbb{R}^n$ orthogonal to the first d rows of $\mathcal{B}$.
- ▶ **Key principle:** If d is not too large, then the shortest vector $\mathbf{v}$ of $\mathcal{L}$ is expected to project to the shortest vector in $\pi_{n-d}(\mathcal{L})$.



Dimensions for Free (D4F)

- ▶ Let $\mathcal{L}$ be a lattice with basis $\mathcal{B}$, and let $\pi_{n-d} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-d}$ be the projection of $\mathbb{R}^n$ orthogonal to the first d rows of $\mathcal{B}$.
- ▶ **Key principle:** If d is not too large, then the shortest vector $\mathbf{v}$ of $\mathcal{L}$ is expected to project to the shortest vector in $\pi_{n-d}(\mathcal{L})$.
- ▶ D4F chooses some large d and calls a sieve on $\pi_{n-d}(\mathcal{B})$. It lifts the shortest vector in $\pi_{n-d}(\mathcal{L})$ to $\mathcal{L}$, and checks if that vector contains the secret key.



Dimensions for Free (D4F)

- ▶ Let $\mathcal{L}$ be a lattice with basis $\mathcal{B}$, and let $\pi_{n-d} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-d}$ be the projection of $\mathbb{R}^n$ orthogonal to the first d rows of $\mathcal{B}$.
- ▶ **Key principle:** If d is not too large, then the shortest vector $\mathbf{v}$ of $\mathcal{L}$ is expected to project to the shortest vector in $\pi_{n-d}(\mathcal{L})$.
- ▶ D4F chooses some large d and calls a sieve on $\pi_{n-d}(\mathcal{B})$. It lifts the shortest vector in $\pi_{n-d}(\mathcal{L})$ to $\mathcal{L}$, and checks if that vector contains the secret key.
 - ▶ If the secret key is not found, then we decrease d and try again.



Guessing Bits



UNIVERSITÀ
DEGLI STUDI
DI MILANO



UNIVERSITÄT
DUISBURG
ESSEN
Offen im Denken

Guessing Bits

- ▶ Idea: We want to somehow “expand” the lattice so that most vectors get longer, while still keeping the target vector below its usual bound.



Guessing Bits

- ▶ Idea: We want to somehow “expand” the lattice so that most vectors get longer, while still keeping the target vector below its usual bound.
- ▶ We do this by guessing c -many bits of either the secret key or of the nonces of the ECDSA setup, in a way which allows us to alter the usual construction of the lattice basis.



Guessing Bits

- ▶ Idea: We want to somehow “expand” the lattice so that most vectors get longer, while still keeping the target vector below its usual bound.
- ▶ We do this by guessing c -many bits of either the secret key or of the nonces of the ECDSA setup, in a way which allows us to alter the usual construction of the lattice basis.
- ▶ This new basis is cleverly constructed so that the new target vector is just as short as the old one, but so that the determinant of the new lattice basis increases by a factor of 2^c .



Guessing Bits

- ▶ Idea: We want to somehow “expand” the lattice so that most vectors get longer, while still keeping the target vector below its usual bound.
- ▶ We do this by guessing c -many bits of either the secret key or of the nonces of the ECDSA setup, in a way which allows us to alter the usual construction of the lattice basis.
- ▶ This new basis is cleverly constructed so that the new target vector is just as short as the old one, but so that the determinant of the new lattice basis increases by a factor of 2^c .
- ▶ The **benefit** is that when the bits are guessed correctly, the target vector is more likely to be the shortest in the lattice. The **cost** is that we must guess up to 2^c combinations and call a lattice sieve each time, which is very slow.



D4F + Guessing Bits: Reasoning

The expected number d of dimensions we can get for free while still expecting the target vector $\mathbf{v}$ to project to a sufficiently short vector in $\pi_{n-d}(\mathcal{L})$ to be found by a lattice sieve can be computed by solving for the largest d such that

$$||\pi_{n-d}(\mathbf{v})|| \leq \sqrt{\frac{4}{3}} \cdot \text{gh}(\pi_{n-d}(\mathcal{L})),$$

where $\text{gh}(\mathcal{L})$ predicts the length of the shortest vector, and depends directly on $\det(\mathcal{B})$.

D4F + Guessing Bits: Reasoning

The expected number d of dimensions we can get for free while still expecting the target vector $\mathbf{v}$ to project to a sufficiently short vector in $\pi_{n-d}(\mathcal{L})$ to be found by a lattice sieve can be computed by solving for the largest d such that

$$\|\pi_{n-d}(\mathbf{v})\| \leq \sqrt{\frac{4}{3}} \cdot \text{gh}(\pi_{n-d}(\mathcal{L})),$$

where $\text{gh}(\mathcal{L})$ predicts the length of the shortest vector, and depends directly on $\det(\mathcal{B})$.

So, by using the bit-guessing methods to increase $\det(\mathcal{B})$, we should increase the expected number of dimensions for free, making each call to D4F faster.



Results

$$||\pi_{n-d}(\mathbf{v})|| \leq \sqrt{\frac{4}{3}} \cdot \text{gh}(\pi_{n-d}(\mathcal{L})),$$

I solved for d using both optimistic and pessimistic assumptions on $||\pi_{n-d}(\mathbf{v})||$, and then found the consequent increase in d upon guessing c bits of nonces or of the secret key



Results

$$\|\pi_{n-d}(\mathbf{v})\| \leq \sqrt{\frac{4}{3}} \cdot \text{gh}(\pi_{n-d}(\mathcal{L})),$$

I solved for d using both optimistic and pessimistic assumptions on $\|\pi_{n-d}(\mathbf{v})\|$, and then found the consequent increase in d upon guessing c bits of nonces or of the secret key

Optimistic: $\|\pi_{n-d}(\mathbf{v})\| = \sqrt{\frac{n-d}{n}} \cdot \|\mathbf{v}\|$

Pessimistic: $\|\pi_{n-d}(\mathbf{v})\| = \|\mathbf{v}\|.$

