Lattice Attacks on Pairing-Based Signatures

Thierry Mefenza, Damien Vergnaud

Lattice Attacks on Pairing-Based Signatures

Thierry Mefenza[1,2] and Damien Vergnaud[3,4]

[1] Département d'informatique de l'ENS
École normale supérieure, CNRS, PSL Research University
75005 Paris, France
[2] INRIA
[3] Sorbonne Universités, UPMC, CNRS
LIP6, Équipe Almasty, Paris, France
[4] Institut Universitaire de France, Paris, France

Abstract. Practical implementations of cryptosystems often suffer from critical information leakage through side-channels (such as their power consumption or their electromagnetic emanations). For public-key cryptography on embedded systems, the core operation is usually group exponentiation – or scalar multiplication on elliptic curves – which is a sequence of group operations derived from the private-key that may reveal secret bits to an attacker (on an unprotected implementation).
We present lattice-based polynomial-time (heuristic) algorithms that recover the signer's secret in popular pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the corresponding exponents are known by the attacker. Our techniques relies upon Coppersmith method and apply to all signatures in the so-called *exponent-inversion* framework in the standard security model (*i.e.* Boneh-Boyen and Gentry signatures) as well as in the random oracle model (*i.e.* Sakai-Kasahara signatures).

Keywords. Cryptanalysis; Side-channel attacks; Lattice attacks; Coppersmith's methods; Pairing-based signatures; Boneh-Boyen signatures; Gentry signatures; Modular Inversion Hidden Number Problem.

1 Introduction

Pairing-based signatures. An identity-based encryption (IBE) scheme is a public key encryption scheme in which a user public key is its identity which may be an arbitrary string such as an email address, a phone number or any other identifier and the user private key is generated by a trusted authority called the private-key generator. In their seminal paper proposing the first IBE scheme, Boneh and Franklin [5] mentioned an interesting transform from an IBE scheme to a signature scheme (whose observation was attributed to Naor). The transformation is as follows: the private-key generator public key and secret key correspond to the public key and secret key of the signature scheme and the user private key generation correspond to signatures generation. The well-known short signature scheme proposed by Boneh, Lynn and Shacham [7, 8] can be seen as an application of Naor transformation to Boneh and Franklin IBE [5].

Pairings (or bilinear maps) are powerful mathematical constructs which have been used since 2000 to design numerous complex cryptographic protocols. There are three known pairing-based approaches to design IBE schemes [9]: *full-domain-hash* [5], *commutative-blinding* [3] and *exponent-inversion* [3, 2, 4]. We focus on the latter framework which gives rise to several short signature schemes thanks to Naor transformation.

Embedded devices and side-channel attacks. The pairing-based signature schemes are very well-suited for resource-limited devices since they produce short signatures and their generation involves only one scalar multiplication on an elliptic curve. In the recent years, theoretical attacks against elliptic curves have shown little improvements whereas *side-channel attacks* became a major threat against elliptic curves implementations [19, 20]. These attacks are based on information gained from the physical leakage of a cryptosystem implementation (such as timing information, power consumption or electromagnetic leaks).

For elliptic-curve cryptography, the core operation is scalar multiplication which is usually computed with the binary method: the binary representation of the (secret) exponent is scanned; for the bit-value zero, a point-doubling is computed, whereas a point-doubling and a point-addition are calculated when the bit-value is one. Distinguishing point-doubling from point-addition in power traces can thus reveal the secret exponent. Classical countermeasures to this *simple power analysis* consist of using regular algorithms for scalar multiplication. In the more involved *differential power analysis*, the idea is to guess the secret bit-by-bit, and try to confirm or infirm the guess for each bit thanks to statistical analysis of several power traces. This approach requires that the same secret is used to perform several cryptographic operations but since pairing-based signatures in the exponent-inversion framework use a different exponent for each new signature, they seem immune to differential power analysis.

In [10], Chari, Rao and Rohatgi introduced the so-called *template attacks* which aim at exploiting side-channel information when only a limited number of leakage traces is available. These attacks require that the attacker is able to perform a profiling of the side-channel leakage. Countermeasures against simple power analysis attacks might not prevent such template-based attacks since they exploit data dependent leakages and not only operation dependent leakages. For pairing-based signatures in the exponent-inversion framework, the signature generation consists of a single scalar multiplication of a fixed base point where the exponent depends algebraically on the secret key, the message and some public randomness. Since the base point is fixed, the first bits of these variable exponents that are processed during the signature computation can only lead to a small set of points and we only need to build templates for the points in this (small) set. In this paper, we show that only a small number of bits of several such exponents is sufficient to determine the secret key via lattice attacks. This approach is similar to lattice attacks [17, 24, 25] combined with template attacks [23] that were proposed against the standardized signature scheme DSA and ECDSA.

Contributions of the paper

We consider several pairing-based signature schemes in the exponent-inversion framework. In [26], Sakai and Kasahara presented the first such scheme (whose security was analyzed in the random oracle model by Zhang, Safavi-Naini and Susilo in [27]). Boneh and Boyen [2] then presented the first pairing-based signature whose security can be proven in the standard security model. In 2006, Gentry [14] proposed yet another scheme using the exponent-inversion paradigm, with a tighter security proof than the earlier proposals.

These schemes can be described in a general simplified form as follows. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same prime order p and let g be a generator of $\mathbb{G}$. We suppose that $(\mathbb{G}, \mathbb{G}_T)$ are equipped with an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function. Let $f, g \in \mathbb{Z}_p[X, Y, M, R]$ be two polynomials of degree at most one in X and Y. The key generation picks uniformly at random two integers $(x, y) \in \mathbb{Z}_p$ as the signing secret key and outputs $(g^x, g^y) \in \mathbb{G}^2$ as the public-key. To sign a message $m \in \{0,1\}^*$, the signer picks uniformly at random $r \in \mathbb{Z}_p$, computes

$$\sigma = g^{f(x,y,\mathcal{H}(m),r)/g(x,y,\mathcal{H}(m),r)}$$

and outputs the pair (σ, r) as the signature. The validity of a signature is checked by verifying whether the following equality holds:

$$e(\sigma, g^{g(x,y,\mathcal{H}(m),r)}) = e(g^{f(x,y,\mathcal{H}(m),r)}, g)$$

where the elements $g^{f(x,y,\mathcal{H}(m),r)}$ and $g^{g(x,y,\mathcal{H}(m),r)}$ can be computed publicly from g^x, g^y, m and r. The three schemes use the following specific polynomials:

- **Sakai-Kasahara [26]:** $f(X, Y, M, R) = 1$, $g(X, Y, M, R) = X + M$
- **Boneh-Boyen [2]:** $f(X, Y, M, R) = 1$, $g(X, Y, M, R) = X + M + YR$
- **Gentry [14]:** $f(X, Y, M, R) = Y + R$, $g(X, Y, M, R) = X + M$

We present lattice-based polynomial-time algorithms that recover the signer's secret $(x, y) \in \mathbb{Z}_p^2$ in these pairing-based signatures when used to sign a constant number of messages under the assumption that blocks of consecutive bits of the corresponding exponents $f(x, y, \mathcal{H}(m), r)/g(x, y, \mathcal{H}(m), r)$ modulo p are known by the attacker. We consider known-message attacks and chosen-message attacks (*i.e.* where the attacker is allowed to choose the message m). The method of this paper is heuristic and uses Coppersmith's lattice technique. Let ℓ denote the bit-length of p and N denote the number of unknown blocks of each signing exponent. In a nutshell, we show that one can recover the secret key if the number of consecutive bits of each unknown block is smaller than the following theoretical values:

- **Sakai-Kasahara:** $\ell/2N^2$
- **Boneh-Boyen:** $\ell/2N^2$
- **Gentry:** ℓ/N

provided that the number of signatures is sufficiently large (see the corresponding sections in the paper for more precise bounds). It is interesting to note, that Gentry scheme which provides the best classical security (tight security reduction in the standard security model), is the weakest against our class of attacks.

More generally, our lattice-based algorithms can be seen as methods to solve variants of the *modular inversion hidden number problem* which was introduced by Boneh, Halevi and Howgrave-Graham in 2001 [6]. This problem is to find a hidden number given several integers and partial bits of the corresponding modular inverse integers of the sums of the known integers and that unknown integer. It was used in [6] to built a pseudo-random number generator and a message authentication code scheme. In [22], the authors mentioned that it is interesting to study a general problem of recovering of an unknown rational function. One can see our results as a first step towards solving this problem.

The efficiency of our (heuristic) attacks has been validated experimentally.

2 Coppersmith Method

We provide a short description of the Coppersmith method [12, 11] for finding small roots of a multivariate modular polynomial system of equations modulo an integer p. We refer the reader to [18] for details and proofs.

Problem definition. Let $f_1(y_1, \dots, y_n), \dots, f_s(y_1, \dots, y_n)$ be irreducible multivariate polynomials defined over $\mathbb{Z}$, having a root $(x_1, \dots, x_n)$ modulo a known integer p namely for $i \in \{1, \dots, s\}$, we have $f_i(x_1, \dots, x_n) \equiv 0 \mod p$. Our goal is to recover the desired root $(x_1, \dots, x_n)$. This problem is generally intractable but becomes solvable (under some conditions) in polynomial time $\log(p)^{O(1)}$ (for constant n and constant total degree of the input polynomials) if the root $(x_1, \dots, x_n)$ is upper-bounded by some values $(X_1, \dots, X_n)$ that depends on p and the degree of the polynomials $f_1, \dots, f_s$.

Polynomials collection. In a first step, one generates a larger collection $\mathfrak{P}$ of polynomials $\{\tilde{f}_1, \dots, \tilde{f}_r\}$ linearly independent having $(x_1, \dots, x_n)$ as a root modulo p^m, for some positive integer m. Usually, the technique consists in taking product of powers of the modulus p, the polynomials f_i for $i \in \{1, \dots, s\}$ and some well-chosen monomials, such as

$$\tilde{f}_\ell = p^{m - \sum_{j=1}^{s} k_{j,\ell}} y_1^{\alpha_{1,\ell}} \cdots y_n^{\alpha_{n,\ell}} f_1^{k_{1,\ell}} \cdots f_s^{k_{s,\ell}}$$

for some positive integers $\alpha_{1,\ell}, \dots, \alpha_{n,\ell}, k_{1,\ell}, k_{s,\ell}$. These polynomials satisfy $\tilde{f}_\ell(x_1, \dots, x_n) \equiv 0 \mod p^m$.

Lattice construction. In a second step, one denotes as $\mathfrak{M}$ the set of monomials appearing in collection of polynomials $\mathfrak{P}$, and one writes the polynomials $\tilde{f}_i(y_1X_1, \dots, y_nX_n)$ for $i \in \{1, \dots, r\}$ as a vector $b_i \in (\mathbb{Z})^\omega$, where $\omega = \sharp\mathfrak{M}$. One then constructs a lattice $\mathcal{L}$ generated by the vectors $b_1, \ldots, b_r$ and computes its reduced basis using the LLL algorithm [21].

Lemma 1. *Let $\mathcal{L}$ be a lattice of dimension ω. In polynomial time, the LLL algorithm given as input of basis of $\mathcal{L}$ outputs a* reduced *basis of $\mathcal{L}$ formed by vectors v_i, $1 \leqslant i \leqslant \omega$ that satisfy:*

$$\|v_1\| \leqslant \|v_2\| \leqslant \ldots \leqslant \|v_2\| \leqslant 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(L)^{\frac{1}{\omega+1-i}}.$$

Generating new polynomials. In a third step of the method, one combines Lemma 2 below (from [16]) and Lemma 1 to obtain n multivariate polynomials $g_1(y_1, \ldots, y_n), \ldots, g_n(y_1, \ldots, y_n)$ having $(x_1, \ldots, x_n)$ as a root over the integers.

Lemma 2. ***(Howgrave-Graham)*** *Let $h(y_1, \ldots, y_n)$ be a polynomial over $\mathbb{Z}$ having at most ω monomials. Suppose that:*

1. *$h(x_1, \ldots, x_n) = 0 \bmod W$ for some $|x_1| < X_1, \ldots, |x_n| < X_n$ and,*
2. *$\|h(X_1 y_1, \ldots, X_n y_n)\| \leqslant \frac{W}{\sqrt{\omega}}$. Then $h(x_1, \ldots, x_n) = 0$ holds over the integers.*

The LLL algorithm run on the lattice $\mathcal{L}$ to obtain n reduced vectors v_i, $i \in \{1, \ldots, n\}$ that we see as some polynomials $\tilde{h}_i(y_1 X_1, \ldots, y_n X_n)$, $i \in \{1, \ldots, n\}$. One can see that for $i \in \{1, \ldots, n\}$, $\tilde{h}_i(x_1, \ldots, x_n) = 0 \mod p^m$, since $\tilde{h}_i$ is a linear combination of $\tilde{f}_1, \ldots, \tilde{f}_r$. Then if the following condition holds:

$$2^{\frac{r(r-1)}{4(r+1-n)}} \det(L)^{\frac{1}{r+1-n}} < \frac{p^m}{\sqrt{\omega}},$$

by Lemmas 1 and 2, $\tilde{h}_i(x_1, \ldots, x_n) = 0$, $i \in \{1, \ldots, n\}$ holds over the integers and we then obtain n polynomials having $(x_1, \ldots, x_n)$ as a root over the integers.

Condition. In our attacks, the number of polynomials in the first step is equal to the number of monomials that appears in the collection, so $r = \omega = \sharp\mathfrak{M}$. In the analysis, we let (as usual in this setting) terms that do not depend on p contribute to an error term ε, and the simplified condition becomes:

$$\det(L) < p^{m(\omega+1-n)}.$$

Under the (heuristic) assumption that all created polynomials in the third step define an algebraic variety of dimension 0, the previous system can be solved (e.g., using elimination techniques such as resultant computation or Gröbner basis) and the desired root recovered in polynomial time[5] $\log(p)^{O(1)}$ (for constant n and constant total degree of the input polynomials). In this paper, we assume that these polynomials define an algebraic variety of dimension 0 and we justify the validity of our attacks by computer experiments.

[5] It is well known that the computational complexity of Gröbner basis algorithm may be exponential or even doubly exponential. In our setting, the number of variables and the total total degree of the input polynomials are fixed and the theoretical complexity is polynomial in the field size (and thus in the security parameter).

3 Lattice Attack on Gentry Signatures

3.1 Gentry Signatures

As mentioned in the introduction, Gentry introduced in [14] an IBE scheme without random oracles with short public parameters and tight security reduction in the standard security model. In this paragraph, we describe the signature scheme obtained by applying Naor transformation to Gentry's IBE. The resulting scheme achieves existential unforgeability under chosen-message attacks in the standard security model.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same prime order p (where $p > 2^{2\lambda}$ for a security parameter λ) and let g be a generator of $\mathbb{G}$. We suppose that $(\mathbb{G}, \mathbb{G}_T)$ are equipped with an efficient computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function. Gentry signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $(x, y) \in \mathbb{Z}_p^2$, computes $h_1 = g^x$ and $h_2 = g^y$ and sets $\mathsf{sk} = (x, y)$ and $\mathsf{pk} = (h_1, h_2) \in \mathbb{G}^2$.
- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$, and picks uniformly at random $r \in \mathbb{Z}_p$. It computes the *signing exponent* $\sigma = (y + r)/(x + \mathcal{H}(m)) \bmod p$ and the group element $s = g^\sigma$. The signature is the pair $(r, s) \in Z_p \times \mathbb{G}$.
- **Signature verification.** Given $(r, s) \in \mathbb{Z}_p \times \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(s, h_2 g^r) \stackrel{?}{=} e\left(g, h_1 g^{\mathcal{H}(m)}\right)$$

3.2 Description of the Attack

In this section, we use Coppersmith's methods to attack Gentry's signatures when the attacker learns some blocks of consecutive bits of the signing exponents.

Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n + 2)$ message/signature pairs $(m_i, (r_i, s_i))_{i \in \{0,\ldots,n+1\}}$ as described above (where n does not depend on the security parameter λ). To simplify the notation in the following, instead of the hash values $\mathcal{H}(m_i)$, we assume that the m_i belongs to $\mathbb{Z}_p$ (for $i \in \{0, \ldots, n+1\}$).

We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents σ_i for $i \in \{0, \ldots, n+1\}$ and its goal is to recover the secret keys x and y. From the knowledge of two different signing exponents σ_i and σ_j for integers $i, j \in \{0, \ldots, n+1\}$ with $i \neq j$, the attacker can actually recover the secrets x and y. Its goal is therefore to recover the hidden bits of two σ_i's in order to obtain x and y.

We have $\sigma_i = (y + r_i)/(x + m_i) \bmod p$ for $i \in \{0, \ldots, n+1\}$ which can be rewritten as:

$$\sigma_i(x + m_i) - y - r_i = 0 \bmod p, \quad i \in \{0, \ldots, n+1\}.$$

We consider a chosen-message attack where the attacker uses an arbitrary unique message m for all signatures (*i.e.* $m_i = m$ for all $i \in \{0, \dots, n+1\}$). Eliminating x and y, in the previous equation, we obtain for $a, b, i \in \{0, \dots, n+1\}$ with $0 \leqslant a < b < i \leqslant n+1$:

$$(r_a - r_b)\sigma_i + (r_i - r_a)\sigma_b + (r_b - r_i)\sigma_a = 0 \bmod p$$

Putting $\sigma_i = \sum_{j=1}^{N} x_{i,j} 2^{k_{i,j}} + \gamma_i$, $i \in \{0, \dots, n+1\}$, where γ_i is known to the attacker and $x_{i,j}$, $j \in \{1, \dots, N\}$ are unknown and $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$ and with the choice $a = 0$, $b = 1$, we obtain a polynomial

$$f_i(z_{0,1}, \dots, z_{0,N}, \dots, z_{n+1,1}, \dots, z_{n+1,N})$$

having as root $X_0 = (x_{0,1}, \dots, x_{0,N}, \dots, x_{n+1,1}, \dots, x_{n+1,N})$ modulo p with:

$$f_i = z_{i,N} + \sum_{j=1}^{N-1} a_{i,j} z_{i,j} + \sum_{j=1}^{N} b_{i,j} z_{1,j} + \sum_{j=1}^{N} c_{i,j} z_{0,j} + \gamma_i (r_0 - r_1) + d_i \bmod p$$

for $i \in \{2, \dots, n+1\}$, where

$$\begin{cases} a_{i,j} = 2^{k_{i,j}} / 2^{k_{i,N}} \bmod p \\ b_{i,j} = 2^{k_{1,j}} (r_i - r_0) / ((r_0 - r_1) 2^{k_{i,N}}) \bmod p \\ c_{i,j} = 2^{k_{0,j}} (r_1 - r_i) / ((r_0 - r_1) 2^{k_{i,N}}) \bmod p \\ \quad d_i = (\gamma_i (r_0 - r_1) + \gamma_1 (r_i - r_0) + \gamma_0 (r_1 - r_i)) / ((r_0 - r_1) 2^{k_{i,N}}) \bmod p \end{cases}$$

for $i \in \{2, \dots, n+1\}$ and $j \in \{1, \dots, N\}$.

We consider the following collection of polynomials (parameterized by some integer $m \in \mathbb{N}$ that does not depend on the security parameter λ):

$$\mathfrak{P}_m = \left\{ f_{i_{0,1}, \dots, i_{n+1,1}, i_{0,2}, \dots, i_{n+1,2}, \dots, i_{0,N}, \dots, i_{n+1,N}} \right\},$$

for all vectors of integers $(i_{0,1}, \dots, i_{n+1,1}, i_{0,2}, \dots, i_{n+1,2}, \dots, i_{0,N}, \dots, i_{n+1,N})$ verifying

$$0 \leqslant i_{0,1} + \cdots + i_{n+1,1} + \cdots + i_{0,N}, \dots, i_{n+1,N} \leqslant m$$

and where the polynomial $f_{i_{0,1}, \dots, i_{n+1,1}, i_{0,2}, \dots, i_{n+1,2}, \dots, i_{0,N}, \dots, i_{n+1,N}}$ is defined by:

$$z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N-1}^{i_{0,N-1}} \dots z_{n+1,N-1}^{i_{n+1,N-1}} z_{0,N}^{i_{0,N}} z_{1,N}^{i_{1,N}} f_2^{i_{2,N}} \dots f_{n+1}^{i_{n+1,N}} p^{m - (i_{2,N} + \cdots + i_{n+1,N})}.$$

One can see that $f_{i_{0,1}, \dots, i_{n+1,1}, i_{0,2}, \dots, i_{n+1,2}, \dots, i_{0,N}, \dots, i_{n+1,N}}(X_0) = 0 \bmod p^m$ for all such vector of integers.

If we use for instance the lexicographical monomial order (with $z_{i,j} < z_{i',j'}$ if ($j < j'$ or ($j = j'$ and $i < i'$)) on the set of monomials, we can define an order over the set of polynomials as:

$$f_{i_{0,1}, \dots, i_{n+1,1}, i_{0,2}, \dots, i_{n+1,2}, \dots, i_{0,N}, \dots, i_{n+1,N}} < f_{i'_{0,1}, \dots, i'_{n+1,1}, i'_{0,2}, \dots, i'_{n+1,2}, \dots, i'_{0,N}, \dots, i'_{n+1,N}}$$

if $z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N}^{i_{0,N}} \dots z_{n+1,N}^{i_{n+1,N}} < z_{0,1}^{i'_{0,1}} \dots z_{n+1,1}^{i'_{n+1,1}} \dots z_{0,N}^{i'_{0,N}} \dots z_{n+1,N}^{i'_{n+1,N}}$.

Using this order, we can write $\mathfrak{P}_m = \{\tilde{f}_i,\, i \in \{1, \ldots, \omega\}\}$, with $\tilde{f}_1 < \tilde{f}_2 < \cdots < \tilde{f}_\omega$ where ω is the number of polynomials. Putting $U = 2^{\max_{i,j} \mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1, \ldots, b_\omega$, where for $i \in \{1, \ldots, \omega\}$, b_i is the coefficient vector of the polynomial $\tilde{f}_i(Uz_{0,1}, \ldots, Uz_{n+1,1}, \ldots, Uz_{0,N}, \ldots, Uz_{n+1,N})$.

One can easily verify that the basis matrix is lower triangular and the diagonal elements are $U^a p^{m-(i_{2,N}+\cdots+i_{n+1,N})}$, where the integer a is equal to $i_{0,1} + \cdots + i_{n+1,1} + i_{0,N} + \cdots + i_{n+1,N}$. The number of variables is $N(n+2)$ and the success condition of Coppersmith's method is $\det(\mathcal{L}) < p^{m(\omega - N(n+2))}$, where $\omega = \sum_{\mathbf{i} \in I} 1$ is the dimension of the lattice with

$$I = \{\mathbf{i} = (i_{0,1}, \ldots, i_{0,N}, \ldots, i_{n+1,N}) | 0 \leqslant i_{0,1} + \cdots + i_{n+1,N} \leqslant m\}.$$

We have $\det(\mathfrak{L}) = U^\eta p^{m\omega} p^{-\mu}$ with

$$\mu = \sum_{\mathbf{i} \in I} i_{2,N} + \cdots + i_{n+1,N} \text{ and } \eta = \sum_{\mathbf{i} \in I} i_{0,1} + \cdots + i_{n+1,N}.$$

If m is large, we can neglect the $N(n+2)$ term in Coppersmith success condition and the asymptotic condition becomes:

$$U^\eta < p^\mu.$$

Using analytic combinatorics methods (see for instance [1] and the references therein), one can verify that when m tends to ∞, we have $\eta = N(n+2)\beta(m, N, n)$ and $\mu = n\beta(m, N, n)$, with

$$\beta(m, N, n) = \frac{m^{N(n+2)+1}}{(N(n+2)+1)!} + o(m^{N(n+2)+1}).$$

Therefore, the attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i for $i \in \{0, \ldots, n+1\}$ satisfies:

$$U < p^{\frac{n}{(n+2)N}} \underset{n \to \infty}{\longrightarrow} p^{\frac{1}{N}}.$$

We can thus heuristically recover (using large[6] constant parameters n and m) the secret key (x, y) if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / N$.

3.3 Experimental Results

We have implemented the attack in Sage 7.6 on a MacBook Air laptop computer (2,2 GHz Intel Core i7, 4 Gb RAM 1600 MHz DDR3, Mac OSX 10.10.5). Table 1 lists the theoretical bound $\delta_{\text{theo}} = \frac{n}{(n+2)N}$ and an experimental bound $\delta_{\exp}$ for a 512-bit prime p (corresponding to a 256-bit security level) with $(n+2)$ signatures (for $n \in \{1, 3, 5\}$) and a few number of unknown blocks ($N \leqslant 2$). We consider the family of polynomials $\mathfrak{P}_m$ with $m = 4$ and $m = 2$. We ran 2^7

N	n	δ_{theo}	δ_{exp}	dimension	m	LLL time(s)	Gröbner basis time(s)
1	1	0.333	0.32	35	4	3.804	4.603
1	3	0.6	0.49	21	2	0.250	0.699
1	5	0.714	0.49	36	2	0.871	38.374
2	1	0.166	0.16	28	2	1.438	0.650
2	5	0.33	0.29	91	2	191.906	556.715

Table 1. Lattice Attack on Gentry signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

experiments for all parameters and Table 1 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

We denote α the maximum number of least significant bits that the attacker knows in each signature σ_j, for all $j \neq 0$ (for instance $\alpha = 0$ means that it does not know any least significant bits of the signatures σ_j, for all $j \in \{1, \dots, n+1\}$). If we know at least $\delta_{\text{exp}}\lceil\log_2(p)\rceil + \alpha$ least significant bits of the signature σ_0 then the Gröbner basis always gives us a system of dimension 0 and we are able to find the N unknown block of sizes $p^{\delta_{\text{exp}}}$ in each signature σ_i for $i \in \{0, \dots, n+1\}$. Otherwise, Gröbner basis computations gives us a system of dimension 1 and we are *a priori* unable to find the unknown blocks (though it is possible in some cases to obtain additional information). This system of dimension 1 occurs because the constructed system admits a large number of "small" solutions. We give an example of this in Appendix A. However, If the condition mentioned above is satisfied, we obtain for $N = 1$ and $n + 2 = 3$, the success rates given in Table 2 (over 250 attacks performed for each parameter pair (m, δ_{exp})).

	$m = 2$	$m = 3$	$m = 4$
$\delta_{\text{exp}} = 0.3225$	100	100	100
$\delta_{\text{exp}} = 0.3250$	98.4	98.4	99.2
$\delta_{\text{exp}} = 0.3275$	90.4	92.8	94.4
$\delta_{\text{exp}} = 0.3300$	66.0	65.2	72.8
$\delta_{\text{exp}} = 0.3325$	10.0	15.2	17.2
$\delta_{\text{exp}} = 0.3350$	0	0	0

Table 2. Lattice Attack on Gentry signatures. Success rates (over 250 attacks performed for each parameter pair (m, δ_{exp})).

[6] In order to reach this asymptotic bound, the constructed matrix is of huge dimension and the resulting polynomial system has a very large number of variables and the computation which is theoretically polynomial-time becomes in practice prohibitive.

4 Lattice Attack on Boneh-Boyen Signatures

4.1 Boneh-Boyen Signatures

Two years before the proposal of Gentry's IBE, Boneh and Boyen proposed two IBE schemes in [2] and described one signature scheme obtained using the Naor transformation in [3]. Their scheme has comparable efficiency properties and also achieves existential unforgeability under chosen-message attacks in the standard security model.

With the same notation as above, Boneh-Boyen signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $(x, y) \in \mathbb{Z}_p^2$, computes $h_1 = g^x$ and $h_2 = g^y$ and sets $\mathsf{sk} = (x, y)$ and $\mathsf{pk} = (h_1, h_2) \in \mathbb{G}^2$.
- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$, and picks uniformly at random $r \in \mathbb{Z}_p$. It computes the *signing exponent* $s = 1/(x + \mathcal{H}(m) + yr) \bmod p$ and the group element $\sigma = g^s$. The signature is the pair $(r, \sigma) \in Z_p \times \mathbb{G}$.
- **Signature verification.** Given $(r, \sigma) \in \mathbb{Z}_p \times \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(\sigma, h_1 \cdot g^{\mathcal{H}(m)} \cdot h_2^r) \overset{?}{=} e\,(g, g)$$

4.2 Description of the Attack

In this section, we use the Coppersmith's methods to attack Boneh-Boyen's signature. Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n+2)$ message/signature pairs $(m_i, (r_i, s_i))_{i\in\{0,\ldots,n+1\}}$ as described above (where n does not depend on the security parameter λ). As above, to simplify the notation, we replace $\mathcal{H}(m_i)$ by $m_i \in \mathbb{Z}_p$ (for $i \in \{0, \ldots, n+1\}$). We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents $\sigma_i = 1/(x + m + yr_i) \bmod p$, for $i \in \{0, \ldots, n\}$, where p, r_i and m_i are known to the attacker and x and y are kept secret.
As for Gentry signatures, from the knowledge of two different signing exponents, the attacker can actually recover the secrets x and y and its goal is to recover the hidden bits of two σ_i's in order to recover x and y.

We have $\sigma_i = 1/(x + m_i + yr_i) \bmod p$ for $i \in \{0, \ldots, n+1\}$ and we have:

$$x + m_i + yr_i - \frac{1}{\sigma_i} = 0 \bmod p, \quad i \in \{0, \ldots, n+1\}.$$

Eliminating x and y and assuming again that the attacker chooses a unique message m (namely $m_i = m$, for all $i \in \{0, \ldots, n+1\}$), we obtain, for $a, b, i \in \{0, \ldots, n+1\}$ with $0 \leqslant a < b < i \leqslant n+1$:

$$(r_b - r_i)\sigma_i\sigma_b + (r_i - r_a)\sigma_i\sigma_a + (r_a - r_b)\sigma_a\sigma_b = 0 \bmod p.$$

Putting $\sigma_i = \sum_{j=1}^{N} x_{i,j} 2^{k_{i,j}} + \gamma_i$, $i \in \{0, \dots, n+1\}$, where γ_i is known to the attacker and $x_{i,j}$, $j \in \{1, \dots, N\}$ are unknown with $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$ and $a = 0$, we obtain a polynomial $f_{0,b,i}(z_{0,1}, \dots, z_{0,N}, \dots, z_{n+1,1}, \dots, z_{n+1,N})$ having as "small" root $X_0 = (x_{0,1}, \dots, x_{0,N}, \dots, x_{n+1,1}, \dots, x_{n+1,N})$ modulo p, where :

$$\begin{aligned} f_{0,b,i} = & \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{b,i,j,k} z_{i,j} z_{b,k} + \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{0,i,j,k} z_{i,j} z_{0,k} + \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{0,b,j,k} z_{b,j} z_{0,k} \\ & + \sum_{j=1}^{N} \alpha_{0,b,i,j} z_{i,j} + \sum_{j=1}^{N} \beta_{0,b,i,j} z_{b,j} + \sum_{j=1}^{N} \gamma_{0,b,i,j} z_{0,j} + \delta_{0,b,i} \bmod p \end{aligned}$$

for $b, i \in \{1, \dots, n+1\}$, $b < i$ and with known coefficients , where $\alpha_{b,i,N,N} = 1$. The set of monomials appearing in the polynomials $f_{0,b,i}$ is:

$$\mathfrak{M} = \left\{ 1, z_{a,j} z_{b,k}, z_{i,j} : i \in \{0, \dots, n+1\} \middle| \begin{array}{l} a, b \in \{0, \dots, n+1\}; a < b \\ j, k \in \{0, \dots, N\} \end{array} \right\}.$$

We consider the following set of polynomials:

$$\mathfrak{P} = \{p\tilde{m}, \tilde{m} \in \mathfrak{M}_1\} \cup \{f_{0,b,i} : b, i \in \{1, \dots, n+1\}; b < i\},$$

where $\mathfrak{M}_1 = \mathfrak{M} \backslash \mathfrak{M}_2$ with $\mathfrak{M}_2 = \{z_{b,N} z_{i,N} : b, i \in \{1, \dots, n+1\}; b < i\}$. One can see that for any polynomial $\tilde{f} \in \mathfrak{P}$, $\tilde{f}(X_0) = 0 \bmod p$. We can define an order on the set of monomials such that all the monomials in $\mathfrak{M}_1$ are smaller than any monomial in $\mathfrak{M}_2$ and for $z_{b,N} z_{i,N}, z_{b',N} z_{i',N} \in \mathfrak{M}_2$, $z_{b,N} z_{i,N} < z_{b',N} z_{i',N}$ if $(b < b'$ or $(b = b'$ and $i < i'))$.

Using that order, we can order the set of polynomials from the smallest element to the greatest as follows:

$$\begin{aligned} \mathfrak{P} &= \{p\tilde{m}_1, \dots, p\tilde{m}_{\omega_1}, f_{0,1,2}, \dots, f_{0,1,n+1}, f_{0,2,3}, \dots, f_{0,2,n+1}, \dots, f_{0,n,n+1}\} \\ &= \{\tilde{f}_1, \dots, \tilde{f}_\omega\} \end{aligned}$$

where $\tilde{m}_1 < \dots < \tilde{m}_{\omega_1}$, ω_1 is the cardinality of $\mathfrak{M}_1$ and ω is the cardinality of $\mathfrak{M}$.

Putting $U = 2^{\max_{i,j} \mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1, \dots, b_\omega$, where for each $i \in \{1, \dots, \omega\}$, b_i is the coefficient vector of the polynomial

$$\tilde{f}_i(Uz_{0,1}, \dots, Uz_{0,N}, \dots, Uz_{n+1,1}, \dots, Uz_{n+1,N}).$$

One can verify that the basis matrix is lower triangular. The number of variables is $N(n+2)$ and the success condition for the Coppersmith's method is:

$$\det(\mathcal{L}) < p^{\omega - N(n+2)+1}, \text{ with } \omega = \sharp\mathfrak{M} = N^2 \frac{(n+1)(n+2)}{2} + (n+2)N + 1.$$

We have $\det(\mathcal{L}) = U^{2N^2 \frac{(n+1)(n+2)}{2} + (n+2)N} p^{\omega - \frac{n(n+1)}{2}}$ and the success condition becomes:

$$U < p^{\frac{\frac{n(n+1)}{2} - N(n+2)+1}{2N^2 \frac{(n+1)(n+2)}{2} + (n+2)N}}.$$

If n is large and since N is small, we can neglect $-N(n+2)+1$ which contribute to a small error term. So the attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i, $i \in \{0, \dots, n+1\}$ satisfies:

$$U < p^{\frac{n(n+1)}{2N^2(n+1)(n+2)+2(n+2)N}} \underset{n \to \infty}{\to} p^{\frac{1}{2N^2}}.$$

We can thus heuristically recover the secret key if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / (2N^2)$.

4.3 Experimental results

Table 3 lists the theoretical bound $\delta_{\text{theo}} = \frac{n(n+1)}{2N^2(n+1)(n+2)+2(n+2)N}$ and an experimental bound δ_{exp} for a 512-bit prime p with $(n+2)$ signatures for a few values of $n \in \{4, 6, 10\}$ and one or two unknown blocks per signatures.

N	n	δ_{theo}	δ_{exp}	dimension	LLL time(s)	Gröbner basis time(s)
1	4	0.277	0.293	22	0.205	0.048
1	6	0.306	0.31	29	1.961	1.008
1	10	0.382	0.38	79	75.086	39.669
2	4	0.076	0.08	73	9.185	3.078
2	6	0.087	0.09	129	232.698	397.900

Table 3. Lattice Attack on Boneh-Boyen signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

We ran 2^7 experiments for all parameters and in all cases (for the bound δ_{exp}), the assumption that the created polynomials define an algebraic variety of dimension 0 was verified. The constructed system was solved using Gröbner basis and the desired root recovered. Table 3 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation (using the same configuration as above).

5 Lattice Attack on Sakai-Kasahara Signatures

5.1 Sakai-Kasahara Signatures

In [26], Sakai and Kasahara presented the first pairing-based signature scheme in the exponent-inversion framework. Their scheme is very close to Boneh-Boyen signature schemes but produces shorter signatures (at the cost of relying on the random oracle heuristic [27]).

With the same notation as above, Sakai-Kasahara signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $x \in \mathbb{Z}_p$, computes $h = g^x$ and sets $\mathsf{sk} = x$ and $\mathsf{pk} = h \in \mathbb{G}$.
- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$. It computes the *signing exponent* $s = 1/(x+\mathcal{H}(m)) \bmod p$ and the group element $\sigma = g^s$. The signature is the group element $\sigma \in \mathbb{G}$.
- **Signature verification.** Given $\sigma \in \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(\sigma, h \cdot g^{\mathcal{H}(m)}) \overset{?}{=} e\left(g, g\right)$$

We present in the following an attack on this scheme when the attacker learns some blocks of consecutive bits of the signing exponents. This computational problem is related to the Modular Inversion Hidden Number Problem which was introduced in 2001 by Boneh, Halevi and Howgrave-Graham [6]. In this problem, the attacker does not know exactly one block of least significant bits of the signing exponents σ_i while our attack considers the setting where the attacker does not know $N \geqslant 1$ different blocks in each σ_i (for any N).

5.2 Description of the Attack

In this section, we use the Coppersmith's methods to attack Sakai-Kasahara signatures. Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n+1)$ message/signature pairs $(m_i, s_i)_{i \in \{0,\ldots,n+1\}}$ as described above (where n does not depend on the security parameter λ). Again, to simplify the notation, we replace $\mathcal{H}(m_i)$ by $m_i \in \mathbb{Z}_p$ (for $i \in \{0,\ldots,n+1\}$). We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents $\sigma_i = 1/(x+m_i) \bmod p$ for $i \in \{0,\ldots,n\}$ and its goal is to recover x. One can see that from the knowledge of a value σ_i, the attacker can actually recover the hidden number x and it is thus sufficient to recover the hidden bits of a single σ_i's in order to recover x.

We have $\sigma_i = 1/(x+m_i) \bmod p$ for $i \in \{0,\ldots,n\}$ which can be rewritten as:

$$x + m_i - \frac{1}{\sigma_i} = 0 \bmod p, \quad i \in \{0,\ldots,n\}.$$

Eliminating x, we obtain:

$$(m_i - m_a)\sigma_i\sigma_a + \sigma_i - \sigma_a = 0 \bmod p \quad a, i \in \{0,\ldots,n\}, 0 \leqslant a < i \leqslant n.$$

Putting, for $i \in \{0,\ldots,n+1\}$, $\sigma_i = \sum_{j=1}^{N} x_{i,j} 2^{k_{i,j}} + \gamma_i$, where γ_i is known to the attacker and $x_{i,j}$ for $j \in \{1,\ldots,N\}$ are unknown with $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$, we obtain a polynomial $f_{a,i}(z_{0,1},\ldots,z_{0,N},\ldots,z_{n,1},\ldots,z_{n,N})$ having as root $X_0 = (x_{0,1},\ldots,x_{0,N},\ldots,x_{n,1},\ldots,x_{n,N})$ modulo p with:

$$f_{a,i} = \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{a,i,j,k} z_{i,j} z_{a,k} + \sum_{j=1}^{N} \beta_{a,i,j} z_{i,j} + \sum_{j=1}^{N} \gamma_{a,i,j} x_{a,j} + \delta_{a,i} \bmod p$$

for $a, i \in \{0, \ldots, n\}$, $a < i$ and with known coefficients, where $\alpha_{a,i,N,N} = 1$. The set of monomials appearing in the polynomials $f_{a,i}$ is:

$$\mathfrak{M} = \{1, z_{a,j} z_{b,k}, z_{i,j} : i \in \{0, \ldots, n\}; a, b \in \{0, \ldots, n\}; a < b; j, k \in \{1, \ldots, N\}\}\,.$$

We consider the following set of polynomials:

$$\mathfrak{P} = \{p\tilde{m}, \tilde{m} \in \mathfrak{M}_1\} \cup \{f_{a,i} : a, i \in \{0, \ldots, n\}; a < i\}\,,$$

where $\mathfrak{M}_1 = \mathfrak{M} \setminus \mathfrak{M}_2$ with $\mathfrak{M}_2 = \{z_{a,N} z_{i,N} : a, i \in \{0, \ldots, n\}; a < i\}$. One can see that for any polynomial $\tilde{f} \in \mathfrak{P}$, $\tilde{f}(X_0) = 0 \bmod p$. We can define an order on the set of monomials such that all the monomials in $\mathfrak{M}_1$ are smaller than any monomial in $\mathfrak{M}_2$ and for $z_{a,N} z_{i,N}, z_{a',N} z_{i',N} \in \mathfrak{M}_2$, $z_{a,N} z_{i,N} < z_{a',N} z_{i',N}$ if $(a < a'$ or $(a = a'$ and $i < i'))$.

Using that order, we can order the set of polynomials from the smallest element to the greatest as follows:

$$\mathfrak{P} = \{p\tilde{m}_1, \ldots, p\tilde{m}_{\omega_1}, f_{0,1}, \ldots, f_{0,n}, f_{1,2}, \ldots, f_{1,n}, \ldots, f_{n-1,n}\} = \{\tilde{f}_1, \ldots, \tilde{f}_\omega\}$$

where $\tilde{m}_1 < \cdots < \tilde{m}_{\omega_1}$, ω_1 is the cardinality of $\mathfrak{M}_1$ and ω is the cardinality of $\mathfrak{M}$. Putting $U = 2^{\max_{i,j} \mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1, \ldots, b_\omega$, where b_i is the coefficient vector of $\tilde{f}_i(Uz_{0,1}, \ldots, Uz_{0,N}, \ldots, Uz_{n,1}, \ldots, Uz_{n,N})$ for $i \in \{1, \ldots, \omega\}$,. One can easily verify that the basis matrix is lower triangular. The number of variables is $N(n+1)$ and the success condition for the Coppersmith's method is:

$$\det(\mathcal{L}) < p^{\omega - N(n+1)+1},$$

with $\omega = \sharp\mathfrak{M} = N^2 \frac{n(n+1)}{2} + (n+1)N + 1$ and $\det(\mathcal{L}) = U^{2N^2 \frac{n(n+1)}{2} + (n+1)N} p^{\omega - \frac{n(n+1)}{2}}$. The success condition then becomes:

$$U < p^{\frac{\frac{n(n+1)}{2} - N(n+1)+1}{2N^2 \frac{n(n+1)}{2} + (n+1)N}}\,.$$

If n is large and since N is small, we can neglect $-N(n+1)+1$ which contributes to a small error. The attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i, $i \in \{0, \ldots, n\}$ satisfies:

$$U < p^{\frac{n(n+1)}{2N^2 n(n+1)) + 2(n+1)N}} \underset{n \to \infty}{\longrightarrow} p^{\frac{1}{2N^2}}\,.$$

We can heuristically recover the secret key of Sakai-Kasahara signatures if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / (2N^2)$.

5.3 Experimental results

Table 4 gives the theoretical bound $\delta_{\text{theo}} = \frac{n(n+1)}{2N^2 n(n+1)) + 2(n+1)N}$ and an experimental bound δ_{exp} for a 512-bit prime p with $(n+1)$ signatures for a few values of $n \in \{4, 6, 10\}$ and one or two unknown blocks per signatures.

N	n	δ_{theo}	δ_{exp}	dimension	LLL time(s)	Gröbner basis time(s)
1	4	0.4	0.39	16	0.015	0.009
1	6	0.4285	0.425	29	0.934	0.267
1	10	0.4545	0.45	67	5.082	4.247
2	4	0.1111	0.1111	51	0.728	0.292
2	6	0.1153	0.1153	99	15.308	14.482

Table 4. Lattice Attack on Sakai-Kasahara signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

We ran 2^7 experiments for all parameters . As in the attack on Boneh-Boyen signatures, the assumption that the created polynomials define an algebraic variety of dimension 0 was verified (in all cases for the bound δ_{exp}) and the constructed system was solved using Gröbner basis and the desired root recovered. Table 4 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation (using the same configuration as above).

6 Conclusion and Open Questions

We presented lattice-based polynomial-time algorithms that recover the signer's secret in popular pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the corresponding exponents are known by the attacker. This partial information can be obtained in practice easily through side-channels (such as the power consumption or the electromagnetic emanations of the device generating the signature).

In order to prevent the leakage of partial information on the exponent, it is customary to use a probabilistic algorithm to encode the sensitive values such that the cryptographic operations only occur on randomized data. In [13], Coron proposed notably to randomize the exponent and the projective coordinates of the base point. It is an interesting question to extend our attacks in such setting (as it was done recently for ECDSA in [15]). It is also interesting to study the security against side-channel attacks of the pairing-based signatures whose design does not rely on the exponent inversion framework (i.e. based on the full domain hash framework and the commutative blinding framework).

Our attacks are heuristic and it would be very interesting to provide proven versions of them (as it was done in [24, 25] for ECDSA signatures). It is also interesting to study further the attack against Gentry signatures when the unknown blocks of consecutive bits overlap. Finally, it would be nice to improve our attacks on Boneh-Boyen and Sakai-Kasahara signatures.

A Concrete Attack Examples against Gentry signatures

In this section, we present two attack examples on Gentry signatures for a 256-bit prime p with 3 signatures (r_0, σ_0), (r_1, σ_1) and (r_2, σ_2) and one T-bit unknown block in each signature, with $T = \lfloor 0.3 \log_2(p) \rfloor$.

We recall that for $i \in \{0, 1, 2\}$, $\sigma_i = g^{s_i}$ where $s_i = (y + r_i)/(x + m) \bmod p$, x and y are the secret keys and p, m and r_i, $i \in \{0, 1, 2\}$ are public information. In this example, we took the following random values:

- $p =$ `9b814891e89496e776bfeeebcac5c74130862914fe2b928d40c3a88323dcbaaf`
- $m =$ `440f4a9df2936c4aad3856ed0ea5cf3d131ef658fc36c2fa56763373288d5519`
- $x =$ `57a7b0913f5202e31555ec9538ff90f38a5e6c53b359edfe1106c8ee9518029a`
- $y =$ `259b67be7de53e0546860379bc31ab9bb30caf68c314a956a1719e18d4a24ae2`
- $r_0 =$ `75c471becf6a9d86aa5480985a95702617892ba84b7662d6bdf3a3c1931abf3b`
- $r_1 =$ `675e28ffbf96b29365ebda463c3a0a4290a284f9fed9ddd0ccdada587c1f0152`
- $r_2 =$ `7961b0df3f0a286547f25da59a7c2a7c28764f4335a0aa2cd5a72ba2393a6cd3`
- $s_0 =$ `45f185a8ce35c2b95b3e1aef9fc516ec9e840c9a5b6b36c70532b10145790401`
- $s_1 =$ `8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab4aafecae41fca50c713`
- $s_2 =$ `57de02b444bb7716c021d21162c3727ba904ae6e4d44aca2ad9f4406669e8744`

and $T = \lfloor 0.3 \log_2(p) \rfloor = 76$.

In the first case, we suppose that we do not know any least significant bits of each signature and show that we are unable to find the unknown blocks since the Gröbner basis gives us a system of dimension 1.

In the second case, we suppose that we know $T + 2$ least significant bits of σ_0 but do not know any least significant bits of s_1, and s_2. We also suppose that we do not know T intermediate bits of s_0 and we show that in this case we are able to find the unknown blocks since the Gröbner basis gives us a system of dimension 0.

First case

- We can write the signatures as:

$$\begin{aligned}
s_0 &= 2^T \cdot \texttt{45f185a8ce35c2b95b3e1aef9fc516ec9e840c9a5b6b3} + z_0, \\
s_1 &= 2^T \cdot \texttt{8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab} + z_1, \\
s_2 &= 2^T \cdot \texttt{57de02b444bb7716c021d21162c3727ba904ae6e4d44a} + z_2,
\end{aligned}$$

  where the T-bit numbers z_0, z_1 and z_2 are the unknown blocks.
- We get the polynomial $f(y_0, y_1, y_2)$ defined by:

$$\begin{aligned}
&y_2 + \texttt{86acc2de9d15dab4df6a8114243623f246376c1103c29ee97a0dd7490f87eb33}\, y_1 \\
&\quad + \texttt{14d485b34b7ebc3297556dd7a68fa34eea4ebd03fa68f3a3c6b5d13a1454cf7b}\, y_0 \\
&\quad + \texttt{11f10fbe97565b062acfb71c6d98f596de6c1e236edaa9168d891d78d66e8c4a}
\end{aligned}$$

  having as root (z_0, z_1, z_2) modulo p.

- Constructing the lattice with $m = 4$, after the LLL reduction and the Gröbner basis computation, we obtain the system of polynomials

$$\begin{cases} f_1(y_0, y_1, y_2) = y_2 - y_0 - \texttt{5dba86c930521258343} \\ f_2(y_0, y_1, y_2) = y_1 - y_0 + \texttt{21c0667cce17b283cee} \end{cases}$$

  having indeed (z_0, z_1, z_2) as root over the integers. However, the dimension of the system is 1 and then we are *a priori* unable to find the unknown blocks.

Second case

- We can write the signatures as:

$$s_0 = \texttt{36c70532b10145790401} + 2^{79} \cdot z_0 + 2^{79+T} \cdot \texttt{8be30b519c6b8572b67c35df3}$$

$$s_1 = 2^T \cdot \texttt{8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab} + z_1$$

$$s_2 = 2^T \cdot \texttt{57de02b444bb7716c021d21162c3727ba904ae6e4d44a} + z_2$$

  where the T-bit numbers z_0, z_1 and z_2 are the unknown blocks.
- If one proceeds like in the attack, we obtain the polynomial $f(y_0, y_1, y_2)$ defined by

$$\begin{aligned} y_2 &+ \texttt{86acc2de9d15dab4df6a8114243623f246376c1103c29ee97a0dd7490f87eb33}\, y_1 \\ &+ \texttt{78836c7dbcc6bee53ea07b359a07fa111e09607336b452976acd0f0ec2a0c985}\, y_0 \\ &+ \texttt{77b82eec348f27f19cb7a6c1cc895cf7261093b80d067ea4eb7b8da90e1ae306} \end{aligned}$$

  having as root (z_0, z_1, z_2) modulo p.
- Constructing the lattice with $m = 4$, after the LLL reduction and the Gröbner basis computation, one obtains the system of polynomials

$$\begin{cases} f_1(y_0, y_1, y_2) = y_2 - \texttt{ca2ad9f4406669e8744} \\ f_2(y_0, y_1, y_2) = y_1 - \texttt{4aafecae41fca50c713} \\ f_3(y_0, y_1, y_2) = y_0 - \texttt{f8a2dd93d081934b6d6} \end{cases}$$

  having (z_0, z_1, z_2) as root over the integers. The dimension of the system is 0 and one finds readily the unknown blocks.

References

1. Benhamouda, F., Chevalier, C., Thillard, A., Vergnaud, D.: Easing Coppersmith methods using analytic combinatorics: Applications to public-key cryptography with weak pseudorandomness. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 9615, pp. 36–66. Springer, Heidelberg, Germany, Taipei, Taiwan (Mar 6–9, 2016)
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 223–238. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004)

3. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 56–73. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004)
4. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21(2), 149–177 (Apr 2008)
5. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) Advances in Cryptology – CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2001)
6. Boneh, D., Halevi, S., Howgrave-Graham, N.: The modular inversion hidden number problem. In: Boyd, C. (ed.) Advances in Cryptology – ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 36–51. Springer, Heidelberg, Germany, Gold Coast, Australia (Dec 9–13, 2001)
7. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) Advances in Cryptology – ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 514–532. Springer, Heidelberg, Germany, Gold Coast, Australia (Dec 9–13, 2001)
8. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. Journal of Cryptology 17(4), 297–319 (Sep 2004)
9. Boyen, X.: A tapestry of identity-based encryption: practical frameworks compared. IJACT 1(1), 3–21 (2008)
10. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski Jr., B.S., Koç, Çetin Kaya., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2002. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer, Heidelberg, Germany, Redwood Shores, CA, USA (Aug 13–15, 2003)
11. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U.M. (ed.) Advances in Cryptology – EUROCRYPT'96. Lecture Notes in Computer Science, vol. 1070, pp. 178–189. Springer, Heidelberg, Germany, Saragossa, Spain (May 12–16, 1996)
12. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U.M. (ed.) Advances in Cryptology – EUROCRYPT'96. Lecture Notes in Computer Science, vol. 1070, pp. 155–165. Springer, Heidelberg, Germany, Saragossa, Spain (May 12–16, 1996)
13. Coron, J.S.: Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç, Çetin Kaya., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES'99. Lecture Notes in Computer Science, vol. 1717, pp. 292–302. Springer, Heidelberg, Germany, Worcester, Massachusetts, USA (Aug 12–13, 1999)
14. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 445–464. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006)
15. Goudarzi, D., Rivain, M., Vergnaud, D.: Lattice attacks against elliptic-curve signatures with blinded scalar multiplication. In: Avanzi, R., Heys, H. (eds.) Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 9-12, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. to appear. Springer (2017)
16. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) 6th IMA International Conference on Cryptography

and Coding. Lecture Notes in Computer Science, vol. 1355, pp. 131–142. Springer, Heidelberg, Germany, Cirencester, UK (Dec 17–19, 1997)

17. Howgrave-Graham, N., Smart, N.P.: Lattice attacks on digital signature schemes. Des. Codes Cryptography 23(3), 283–290 (2001)
18. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In: Lai, X., Chen, K. (eds.) Advances in Cryptology – ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 267–282. Springer, Heidelberg, Germany, Shanghai, China (Dec 3–7, 2006)
19. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology – CRYPTO'96. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 1996)
20. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology – CRYPTO'99. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999)
21. Lenstra, A.K., Lenstra, H.W.J., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261, 515–534 (1982)
22. Ling, S., Shparlinski, I.E., Steinfeld, R., Wang, H.: On the modular inversion hidden number problem. J. Symb. Comput. 47(4), 358–367 (2012)
23. Mulder, E.D., Hutter, M., Marson, M.E., Pearson, P.: Using Bleichenbacher's solution to the hidden number problem to attack nonce leaks in 384-bit ECDSA. In: Bertoni, G., Coron, J.S. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2013. Lecture Notes in Computer Science, vol. 8086, pp. 435–452. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–23, 2013)
24. Nguyen, P.Q., Shparlinski, I.: The insecurity of the digital signature algorithm with partially known nonces. Journal of Cryptology 15(3), 151–176 (2002)
25. Nguyen, P.Q., Shparlinski, I.E.: The insecurity of the elliptic curve digital signature algorithm with partially known nonces. Des. Codes Cryptography 30(2), 201–217 (2003)
26. Sakai, R., Kasahara, M.: ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054 (2003), http://eprint.iacr.org/2003/054
27. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 2947, pp. 277–290. Springer, Heidelberg, Germany, Singapore (Mar 1–4, 2004)