The Insecurity of Esign in Practical Implementations

Pierre-Alain Fouque*,
Nick Howgrave-Graham**,
Gwenaëlle Martinet*** and
Guillaume Poupard***

* Ecole normale supérieure, France
** NTRU Cryptosystems, USA
*** DCSSI Crypto Lab, France

Our goal

- Power analysis techniques (**SPA**) are very useful tools to detect some « *events* »
- Lattice reduction algorithm (**LLL**) is a very useful tool for « *classical cryptanalysts* »
- this paper : an example of a combination of these techniques applied on Esign

Esign

- Private key: p and q two primes of k-bit length
- Public key: $N = p^2q$ and $e > 4$
- Signature scheme based on the Approximate e-th Root problem: given a modulus $N=p^2q$, an exponent e larger than 4 and $y \in Z^*_N$, find $x \in Z^*_N$ such that $x^e \in [\, y, y + 2^{2k-1} \,]$

Esign description

Signature of a message M

- $H(M)$ hash of M on $k-1$ bits, $y = 0 \,||\, H(M) \,||\, 0^{2k}$
- r randomly chosen in Z^*_{pq}
- $z = y - r^e \bmod N$
- $w_0 = \lceil z / pq \rceil$
- $w_1 = w_0 pq - z$, if $w_1 > 2^{2k-1}$ then choose another r
- $u = w_0 (er^{e-1})^{-1} \bmod p$
- $s = r + upq$

Lattice definition

$$L = \begin{pmatrix} V_1 \\ V_i \\ V_d \end{pmatrix} = \begin{pmatrix} s_1 & K & 0 & 0 \\ s_i & 0 & K & 0 \\ s_d & 0 & 0 & K \end{pmatrix}$$

Lattice = set of all the linear combinations, with integer coefficients, of basis vectors $V_1, \ldots V_d$

$$L = \left\{ \sum_{i=1}^{d} c_i V_i \text{ s.t. } (c_1, c_2, \ldots c_i, \ldots, c_d) \in Z^d \right\}$$

Lattice reduction

- *Lattice reduction* = computation of a basis that generates the same lattice and such that
  - the vectors of the basis are « ***short*** »
  - the vectors of the basis are « ***almost orthogonal*** »
- similar to *Gram-Schmidt* reduction but using integer coefficients

LLL

Lattice reduction algorithm : **LLL**
(Lenstra, Lenstra, Lovasz, 1982)

$\longrightarrow$ Use of **LLL** to find a ***short vector*** in a lattice

$\longrightarrow$ If we know that a lattice has an « *abnormally short vector* », we can use **LLL** as a « *short vector oracle* »

Basic idea

- Suppose we have access to an oracle O_{pq} s.t.:
  - input : $s \in Z_N$, for $N = p^2q$
  - output : MSB_b (s mod pq)
- Such oracle allows to get numbers of the form $s = r + u\,pq$ where $|r| < pq / 2^b$
- Lattice reduction allows to recover p

First Attack

- Let d integers $s_i = r_i + u_i pq$ for $1 \leq i \leq d$ where $r_i < pq / 2^b$ and r_i and u_i are unknown
- Let L be the lattice spanned by the rows of the following matrix :

$$\begin{pmatrix} N & 0 & \cdots & 0 & \cdots & 0 \\ 0 & N & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & & N & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ & & & & & N \\ s_1 & s_2 & & s_i & \cdots & s_d \end{pmatrix}$$

Each vector is of norm N, except the $(d+1)^{th}$ of norm approximatively $\sqrt{d} \times N$

First Attack

For $1 \leq i \leq d$, $p \times s_i = p \times r_i + u_i \times N$. Thus:

$$(-u_1,-u_2,\ldots,-u_i,\ldots,-u_d,-p) \times \begin{pmatrix} N & 0 & \cdots & 0 & \cdots & 0 \\ 0 & N & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & & N & \cdots & 0 \\ \vdots & \vdots & & & \ddots & \vdots \\ & & & & & N \\ s_1 & s_2 & & s_i & \cdots & s_d \end{pmatrix}$$

$$= (p \times r_1, p \times r_2, \ldots, p \times r_i, \ldots, p \times r_d) = V$$

Thus V is a vector of the lattice

First attack

$$V = (p \times r_1, p \times r_2, \ldots, p \times r_i, \ldots, p \times r_d)$$

- If b is large enough, the vector V is of small norm compared to the original vectors of the matrix. Indeed,
  - $p \times r_i < N / 2^b$
  - $\|v\| < \sqrt{d} \times N / 2^b$

  $\Rightarrow$ we hope LLL will return V

- We can get the factor p with a simple gcd of the coefficients of V

Efficiency of the attack

- This basic attack requires the knowledge of about the 24 MSBs for each random value used for 64 Esign signatures and a 1152-bit modulus
- We improve the efficiency of the attack by using another lattice. Only 8 bits for 57 signatures are needed

Improvement of the attack

Basic idea: recovering the value u of a signature $s = r + upq$

From this value, we get

$$s / u = r / u + pq = pq + \alpha$$

where α is a k-bit value

Thus, the bits of s / u and pq collide except on their k LSBs

Computing $N / (pq + \alpha)$ gives p

Improvement of the attack

Finding small linear combinations of the s_i using the LLL algorithm on the lattice spanned by the rows of the following matrix, for K of about $N^{2/3}/\varepsilon$

$$L = \begin{pmatrix} s_1 & K & \cdots & 0 & \cdots & 0 \\ \vdots & & \ddots & & & \vdots \\ s_i & 0 & \cdots & K & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ s_d & 0 & \cdots & 0 & \cdots & K \end{pmatrix}$$

Improvement of the attack
How to recover the u_is ?

- Each new vector basis can be written as $(\sum c_i s_i , Kc_1, \ldots , Kc_d)$
- Due to the properties of the LLL algorithm, $\sum c_i s_i$ is about Kc_i which is “small”
- Thus $\sum c_i s_i = \sum c_i r_i + \sum c_i u_i pq$ is also “small” and $\sum c_i u_i = 0$
- Finally LLL gives us a system of equations in the u_i variables

Results on Esign

- Given an oracle that on input s returns the b MSB of (s mod pq), where

$$b \geq \lceil n / (3 \times (d\text{-}1)) + \log d + 1 \rceil$$

then we can factor $N = p^2q$ from d independent numbers $s \in Z_N$

- For Esign, if the b MSBs of the random r are known, we get an integer $s' = r' + upq$ where $r' < pq / 2^b$ and s' can be used to build the lattice

Practical experiments

- For a 1152-bit modulus ($n = 1152$) and 57 signatures ($d = 57$), $b = 11$ bits are needed
- Experimentally, b = 8 bits are enough and the LLL algorithm requires 3 minutes

Remarks

- The attacks can also be mounted if the LSBs of the random are known
- The attack can be transformed for RSA modulus: cf [FMP03] at CHES ’03 against the RSA-CRT with unbalanced modulus

Relation to the security proof

This results are not in contradiction with the security proofs given at Crypto ’02 or at this conference

Here, the attack uses a stronger attacker model where the adversary is able to obtain the LSBs or MSBs of the randoms

Practical consequences

- In Esign implementations, it is important that the PRBG is cryptographically secure
  - Neither SPA attack allowing to recover some bits of the randoms generated
  - Nor design weakneasses allowing to predict these bits

Scheme	Modulus size	Signatures required	Bits required
DSA	160	100	3
Esign	1152	57	8