Lattice-Based Fault Attacks on RSA Signatures

Mehdi Tibouchi

École normale supérieure

Workshop on Applied Cryptography, Singapore, 2010-12-03

Gist of this talk

- Review a classical attack on RSA signatures by fault injection.
- Show how an industry standard for signatures avoids this attack.
- Explain two techniques based on mathematical lattices to extend the attack and make the standard vulerable nonetheless.

Outline

RSA-CRT and the Bellcore Attack

ISO 9796-2 Signatures

Polynomial Attack

Orthogonal Lattice Attack

Outline

RSA-CRT and the Bellcore Attack

ISO 9796-2 Signatures

Polynomial Attack

Orthogonal Lattice Attack

Signing with RSA-CRT

In RSA-based signature schemes, a signer with modulus $N = pq$ and key pair (e, d) signs a message m by computing:

1. $\sigma_p = \mu(m)^d \bmod p$
2. $\sigma_q = \mu(m)^d \bmod q$
3. $\sigma = \mathrm{CRT}(\sigma_p, \sigma_q) \bmod N$

where μ is the encoding function of the scheme.

The Chinese Remainder Theorem offers a welcome 4-fold speed-up in (often costly) signature generation.

The Bellcore fault attack

The problem with CRT: fault attacks. A fault in signature generation makes it possible to recover the secret key:

1. $\sigma_p = \mu(m)^d \bmod p$
2. $\sigma'_q \neq \mu(m)^d \bmod q \quad \leftarrow$ fault
3. $\sigma' = \mathrm{CRT}(\sigma_p, \sigma'_q) \bmod N \quad \leftarrow$ faulty signature

Then ${\sigma'}^e$ is $\mu(m) \bmod p$ but not mod q, so the attacker can then factor N:

$$p = \gcd({\sigma'}^e - \mu(m), N)$$

This attack applies to:

- any deterministic padding; e.g. FDH, $\sigma = H(m)^d \bmod N$
- any probabilistic padding with public randomizer; e.g. PFDH, $\sigma = \big(r, H(m\|r)^d \bmod N\big)$

Fault attacker's deadlock

The Bellcore attacks does not apply when only a part of the signed encoding is known to the attacker.
Examples:

- $\sigma = (m \| r)^d \bmod N$, where r is a large enough random nonce unknown to the attacker.
- $\sigma = \big(\omega \| G_1(\omega) \oplus r \| G_2(\omega)\big)^d \bmod N$, where r is a random nonce and $\omega = H(m\|r)$. This is PSS.

The attacker doesn't know r, cannot compute $\sigma' - \mu(m)$ to factor N: the Bellcore attack is thwarted (unless r is short enough for exhaustive search).

Outline

RSA-CRT and the Bellcore Attack

ISO 9796-2 Signatures

Polynomial Attack

Orthogonal Lattice Attack

ISO 9796-2

- ISO/IEC 9796-2 is an international standard for RSA signatures, with large industry adoption (especially in banking: EMV).
- It defines signature paddings with partial message recovery, i.e. a portion of the message is recovered as part of signature verification.
- In particular, not all of the message is available to an attacker: fault attacker's deadlock!
- However, with lattice-based techniques, one can extend the Bellcore attack to the traditional ISO/IEC 9796-2 padding.
- Newer versions of the standard also include a variant of PSS, which is secure against random faults [CM09], but it has yet to gain momentum in applications.

The classical ISO 9796-2 padding

- Messages m are divided as $m[1]\|m[2]$, and only $m[2]$ is transmitted; $m[1]$ is recovered during signature verification. More precisely:

$$\mu(m) = \texttt{6A}_{16}\|m[1]\|H(m)\|\texttt{BC}_{16}$$

- In cases of interest (*e.g.* EMV signatures), we can write:

$$m[1] = \alpha\|r\|\alpha' \qquad m[2] = \textsc{data}$$

  where α, α' are known bit patterns, and r is unknown. $\textsc{data}$ is a string that may or may not be known to the attacker.

- The encoded message is thus:

$$\mu(m) = \texttt{6A}_{16}\|\alpha\|r\|\alpha'\|H(\alpha\|r\|\alpha'\|\textsc{data})\|\texttt{BC}_{16}$$

  where the highlighted parts are unknown.

- In the current version of the standard, H has to be a hash function of $k_h \geq 160$ bits (in practice: SHA-1).
- The total number of unknown bits is $k_r + k_h$.

Outline

RSA-CRT and the Bellcore Attack

ISO 9796-2 Signatures

Polynomial Attack

Orthogonal Lattice Attack

The affine equation

- The encoded message

  $$\mu(m) = \mathtt{6A}_{16} \| m[1] \| H(m) \| \mathtt{BC}_{16}$$

  can be written:

  $$\mu(m) = t + r \cdot 2^{n_r} + H(m) \cdot 2^8$$

  where t is a known value, both r and $H(m)$ are unknown.
- A faulty signature σ' yields an equation of the form:

  $$A + B \cdot r + C \cdot H(m) \equiv 0 \pmod p$$

  with $A = t - \sigma'^e$, $B = 2^{n_r}$, $C = 2^8$.
- Hence $(r, H(m))$ is a root mod p of the bivariate polynomial $A + Bx + Cy$.

Coppersmith-like trick

Now we are left with solving the affine equation

$$A + Bx + Cy = 0 \mod p$$

which admits a "small" root $(x_0, y_0) = (r, H(m))$. However p is unknown.
At CHES 2009, Coron et al. [CJKNP09] proposed the following attack:

- Apply the method of Herrmann and May [HM08].
- The method is based on the Coppersmith's technique for finding small roots of polynomial equations.
- Coppersmith technique uses lattice reduction to obtain (x_0, y_0).
- Finally, given (x_0, y_0) we can compute $\mu(m)$ and factor N since $p = \gcd(\sigma'^e - \mu(m), N)$.

Bounds on UMP size

For a balanced RSA modulus, it follows from the computations in [HM08] that the attack works when the "small root" (x_0, y_0) satisfies $|x_0| < N^\gamma$ and $|y_0| < N^\delta$ where

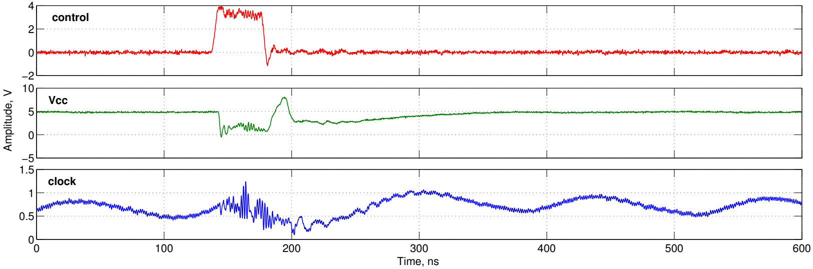
$$\gamma + \delta \leq \frac{\sqrt{2} - 1}{2} \cong 0.207$$

This means that for a 1024-bit RSA modulus N, the combined size of unknown message parts (UMPs) x_0 and y_0 can be at most 212 bits.

Applied to our context, this implies that for ISO/IEC 9796-2 with $k_h = 160$, the size of r can be as large as 52 bits.

Physical fault injection

- 1536-bit RSA with CRT on ATmega128: running time several minutes at 7.68 MHz.
- Spike attack: 40 ns cut-off in power supply (using FPGA).



Limitations of this attack

- Severe size constraint on $r, H(m)$: in practical settings (e.g. EMV), r is often significantly longer than 52 bits.
- The size of the lattice to be reduced grows when the UMP size gets close to the limit: even the small theoretical maximum is difficult to reach in practice.
- To handle larger UMPs, up to $0.5 \cdot n$ in theory, one can take advantage of multiple faults.
- However, complexity grows exponentially with the number of faulty signatures. Going beyond about $0.23 \cdot n$ is unfeasible in practice.

Outline

RSA-CRT and the Bellcore Attack

ISO 9796-2 Signatures

Polynomial Attack

Orthogonal Lattice Attack

A better way to use more faults

When several faulty signatures are available, it is in principle possible to extend the polynomial attack to take advantage of them for dealing with larger UMPs, but the lattice dimensions quickly become unmanageable.

At CT-RSA 2010, Coron, Naccache and T. presented another multiple fault attack on ISO 9796-2 based on different principle, and that lifts most limitations of the CJKNP attack:

- Simpler and purely linear: doesn't suffer from algebraic independence problems of multivariate Coppersmith techniques.
- Scales well with the number of faults: lattice dimension is linear in the number of faults; easy to handle UMPs almost as large as the theoretical maximum.
- Applicable to many EMV signature formats.

Orthogonal lattice attack rundown

Recall that each faulty ISO 9796-2 signature σ'_i gives an equation $A_i + Bx_i + Cy_i \equiv 0 \pmod{p}$, with $(x_i, y_i) = (r_i, H(m_i))$. Dividing by B, we get affine relations:

$$a_i + x_i + cy_i \equiv 0 \pmod{p} \qquad (*)$$

Given ℓ faulty signatures, the attack proceeds as follows:

1. Linearize: using standard lattice reduction technique la Nguyen-Stern, find vectors $\mathbf{u_j} = (u_{1j}, \ldots, u_{\ell j})$ such that $\mathbf{u_j} \cdot \mathbf{a} \equiv 0 \pmod{N}$. Use them to cancel constant terms between the relations $(*)$.
2. Orthogonalize: if the vectors as small enough, each $\mathbf{u_j}$ is orthogonal to $\mathbf{x}$ and $\mathbf{y}$. Deduce a $\mathbb{Z}$-lattice containing $\mathbf{x}$ and $\mathbf{y}$.
3. Factor: find a vector $\mathbf{v}$ orthogonal to both $\mathbf{x}$ and $\mathbf{y}$ mod N, but not to $\mathbf{a}$. Then $p = \gcd(\mathbf{v} \cdot \mathbf{a}, N)$.

Size constraints

For the attack to work, we need the $\mathbf{u_j}$ from the previous slide to be "short enough." How short is short enough?

Heuristically, the shortest vector in the lattice

$$L(c,p) = \{(\alpha,\beta) \in \mathbb{Z}^2 : \alpha + c \cdot \beta = 0 \mod p\}$$

is of length $\approx \sqrt{p}$. Thus, if $|\mathbf{u_j} \cdot \mathbf{x}| \cdot |\mathbf{u_j} \cdot \mathbf{y}| < p \approx N^{1/2}$, we expect the attack to work.

Let N^γ and N^δ be the bounds on x_i and y_i. The LLL-reduced vectors $\mathbf{u_j}$ have components smaller than about $N^{1/\ell}$, so:

$$|\mathbf{u_j} \cdot \mathbf{x}| \lesssim N^{1/\ell+\gamma} \quad |\mathbf{u_j} \cdot \mathbf{y}| \lesssim N^{1/\ell+\delta}$$

Hence the heuristic size constraint:

$$\frac{2}{\ell} + \gamma + \delta < \frac{1}{2}$$

Verifying the size constraint

For $\gamma + \delta = 1/3$, the previous heuristic argument predicts that 13 faults are needed to factor *N*. Very well verified in practice, both for balanced and unbalanced γ, δ.

Number of faults ℓ	12	13	14
Success rate with $\gamma = \delta = \frac{1}{6}$	13%	100%	100%
Success rate with $\gamma = \frac{1}{4}$, $\delta = \frac{1}{12}$	0%	100%	100%
Average CPU time (seconds)	0.19	0.14	0.17

Efficiency of both attacks

Number of required faults, lattice dimension and CPU time for various UMP sizes, in the orthogonal lattice attack (left) and the polynomial attack (right).

$\gamma + \delta$	ℓ_{orth}	ω_{orth}	CPU time	ℓ_{pol}	ω_{old}	CPU time
0.204	7	8	0.03 s	3	84	49 s
0.214	8	9	0.04 s	2	126	22 min
0.230	8	9	0.04 s	2	462	centuries?
0.280	10	11	0.07 s	6	6188	—
0.330	14	15	0.17 s	8	2^{21}	—
0.400	25	26	1.44 s	—	—	—
0.450	70	71	36.94 s	—	—	—

The orthogonal lattice attack handles much larger parameters, but always requires a larger number of faults for a given UMP size: the polynomial attack is preferable for very small sizes.

Conclusion

- The Bellcore attack doesn't apply directly to ISO 9796-2 signatures.
- However, lattice reduction techniques can extend the fault attack to this setting in a practical way: given a few faulty ISO 9796-2 signatures, it is fast and easy to factor the public modulus.
- Signature formats based on this standard, such as EMV, are vulnerable.
- In situations where fault attacks are a concern, provably secure encodings, such as PSS, should be prefered. If it is not possible, CRT fault countermeasures like Shamir's are necessary.

Thank you!