RSA-CRT fault attack

How RSA signatures work

1. $p, q \in \mathbb{Z}_{2^{1024}}$ – two sufficiently large primes and modulus $n = pq$
2. private exponent $d = e^{-1} \in \mathbb{Z}_{\varphi(n)}$.
3. If μ is the padding scheme (such as FDH, PFDH, PKCS #1 v1.5, PKCS #1 v2.5), the signature is $\sigma = (\mu(m))^d \pmod{n}$.
4. Verification $\sigma^e \pmod{n} = \mu(m)$.

Calculating in $\mathbb{Z}_n$ (if n is a 2048-bit integer) is slow.

```
$openssl speed rsa2048

... 1431 signatures per second,
... 51952 veritifactions per second
```

At least much slower, compared to calculating in $\mathbb{Z}_p$ and $\mathbb{Z}_q$ separately. This gives 4 times performance increase. The Chinese Remainder Theorem (CRT) allows to speed-up computations.

$$\begin{cases} \sigma_p \equiv m^{d \bmod \varphi(p)} \pmod{p} \\ \sigma_q \equiv m^{d \bmod \varphi(q)} \pmod{q} \end{cases} \implies \sigma = CRT(\sigma_p, \sigma_q) \mod n\ .$$

Bellcore attack

$$\begin{cases} \sigma_p \equiv m^{d \bmod \varphi(p)} \pmod{p} \\ \hat{\sigma}_q \not\equiv m^{d \bmod \varphi(q)} \pmod{q} \end{cases} \implies \hat{\sigma} = CRT(\sigma_p, \hat{\sigma}_q) \mod n\ .$$

If an attacker manages to inject a fault so that she obtains two RSA signatures, one valid signature σ, and an invalid signature $\hat{\sigma}$, then

$$\begin{cases} (\sigma - \hat{\sigma}) \equiv 0 \pmod{p} \\ (\sigma - \hat{\sigma}) \not\equiv 0 \pmod{q} \end{cases} \implies \gcd(\sigma - \hat{\sigma}, n) = p\ .$$

This case can be reduced to just the knowledge of one faulty signature – the Boneh–DeMillo–Lipton attack.

Boneh–DeMillo–Lipton attack

$$\begin{cases} \sigma_p \equiv m^{d \bmod \varphi(p)} \pmod{p} \\ \hat{\sigma}_q \not\equiv m^{d \bmod \varphi(q)} \pmod{q} \end{cases} \implies \hat{\sigma} = CRT(\sigma_p, \hat{\sigma}_q) \mod n\ .$$

The attack is based on the observation that

$$\begin{cases} \hat{\sigma}^e \equiv m \pmod{p} \\ \hat{\sigma}^e \not\equiv m \pmod{q} \end{cases} \implies \begin{cases} \hat{\sigma}^e - m \equiv 0 \pmod{p} \\ \hat{\sigma}^e - m \not\equiv 0 \pmod{q} \end{cases} \implies \begin{cases} p \mid \hat{\sigma}^e - m \\ q \nmid \hat{\sigma}^e - m \end{cases} \implies \gcd(\hat{\sigma}^e - m, n) = p\ .$$

In some cases the attacker knows the message that is signed, i.e. attacks against TLS, where messages have pre–defined format, and the values for the required fields, necessary to reconstruct m, can be obtained by listening over the TLS handshake message exchange. This attack works for any deterministic padding like FDH or PSS.

Seifert attack

The Bellcore and Boneh–DeMillo–Lipton attacks are fault injection attacks targeted against modular exponentiation. The Seifert attack targets modular reduction ($\mod n$) instead.

$$\begin{cases} \sigma_p \equiv m^{d \bmod \varphi(p)} \pmod p \\ \sigma_q \equiv m^{d \bmod \varphi(q)} \pmod q \end{cases} \implies \hat{\sigma} = CRT(\sigma_p, \hat{\sigma}_q) \mod n \ .$$

The attack is executed using orthogonal lattices.