SIEMENS

# Wagner's Attack on a Secure CRT-RSA Algorithm Reconsidered

FDTC 2006


Johannes Blömer
Paderborn University
Institute for Computer Science
Paderborn, Germany

Martin Otto
Siemens AG
Corporate Technology CT IC3
Munich, Germany

SIEMENS

# Content

- The "Secure CRT-RSA Algorithm"
- Wagner’s Attack (random faults)
- Flaw
- Wagner’s Attack (bit and byte version)
- Outlook: fixing the algorithm
- Conclusion

# The BOS-Algorithm

$$S_p := m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1$$
$$S_q := m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2$$
$$S := \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2$$

CRT-RSA, enhanced with two strong primes

$$c_1 := (m - S^{e_{t_1}} + 1) \bmod t_1$$
$$c_2 := (m - S^{e_{t_2}} + 1) \bmod t_2$$

error detection step

$d \cdot e_{t_i} \equiv 1 \bmod \varphi(t_i)$

$$Sig := S^{c_1 \cdot c_2} \bmod N$$

infective computation

- Presented at CCS 2003 by Blömer, Otto, Seifert
- Aims at protecting CRT-RSA against Bellcore-attacks

# How The Protection Is Intended To Works

$$S_p := m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1$$
$$\tilde{S}_q \neq m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2$$
$$S := \mathrm{CRT}(S_p, \tilde{S}_q) \bmod N \cdot t_1 \cdot t_2$$

CRT-RSA, enhanced with two strong primes

$$c_1 := (m - S^{e_{t_1}} + 1) \bmod t_1$$
$$c_2 := (m - S^{e_{t_2}} + 1) \bmod t_2$$

error detection step
$d \cdot e_{t_i} \equiv 1 \bmod \varphi(t_i)$

$$Sig := S^{c_1 \cdot c_2} \bmod N$$

infective computation

- If no error occurred: $c_1 = (m - m^{(d \cdot e_{t_1})}) + 1 \equiv 1 \bmod t_1$
- If an error occurs: $c_2 = (m - \tilde{m}) + 1 \not\equiv 1 \bmod t_2$ w.h.p.
- Final exponentiation: $Sig = S^{c_1 \cdot c_2} \bmod N$

# Wagner's Attack (random fault version)

- What is a random fault?
  - $x \overset{\lightning}{\mapsto} x + e(x)$, where $e(x) \in_R [-x, 2^{l(x)} - x - 1]$
- Special random fault as proposed by Wagner:
  - Faults do not change the 160 most significant bits
  - number 160 motivated by $l(t_1) = l(t_2) = 80$
- Attack aims on $N$ in last line:

$$Sig = S^{c_1 c_2} \bmod N$$

- Attack aims at disclosing $t_1$ and $t_2$
- Fact: Knowledge of $t_1$ and $t_2$ breaks the BOS-scheme
  - For details, see the original CCS'03 paper

# Wagner's Attack: Exploiting a Faulty $N$ (1)

$$
\begin{aligned}
S_p &:= m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1 \\
S_q &:= m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2 \\
S &:= \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2 \\
c_1 &:= (m - S^{e_{t_1}} + 1) \bmod t_1 \\
c_2 &:= (m - S^{e_{t_2}} + 1) \bmod t_2 \\
Sig &:= S^{c_1 \cdot c_2} \bmod N
\end{aligned}
$$

} Attack targets $N$

1. Gather a correct final result $Sig = m^d \bmod N$
2. Re-run on same input, attack $N$ in last line
3. Gather a faulty final result $\widetilde{Sig} = m^d \bmod \tilde{N}$

# Wagner's Attack: Exploiting a Faulty $N$ (2)

$$S_p := m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1$$
$$S_q := m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2$$
$$S := \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2$$
$$c_1 := (m - S^{e_{t_1}} + 1) \bmod t_1$$
$$c_2 := (m - S^{e_{t_2}} + 1) \bmod t_2$$
$$Sig := S^{c_1 \cdot c_2} \bmod N$$

$S = (S \bmod N) + k \cdot N$, with $0 \le k < 2^{161}$, if $l(t_1 \cdot t_2) \le 2^{160}$

Attack targets $N$ ($c_1 \cdot c_2 = 1$)

4. $\widetilde{Sig} - Sig \equiv S - Sig \equiv S - (S - k \cdot N) \bmod \tilde{N}$
$\equiv k \cdot (N - \tilde{N}) \bmod \tilde{N}$

5. Analysis shows: w.h.p. we have that $\left|\widetilde{Sig} - Sig\right| = x \cdot k$

# Wagner's Attack: Exploiting a Faulty $N$ (3)

$$S_p := m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1$$
$$S_q := m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2$$
$$S := \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2$$
$$c_1 := (m - S^{e_{t_1}} + 1) \bmod t_1$$
$$c_2 := (m - S^{e_{t_2}} + 1) \bmod t_2$$
$$Sig := S^{c_1 \cdot c_2} \bmod N$$

$S = (S \bmod N) + k \cdot N$, with $0 \le k < 2^{161}$, if $l(t_1 \cdot t_2) \le 2^{160}$

Attack targets $N$ ($c_1 \cdot c_2 = 1$)

5. Analysis shows: w.h.p. we have that $\left|\widetilde{Sig} - Sig\right| = x \cdot k$
6. Several attacks and gcd computations reveal $k$
7. Still, we wish to learn $t_1$ or $t_2$ . . .

# Wagner's Attack: The Flaw

$$
\begin{aligned}
S_p &:= m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1 \\
S_q &:= m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2 \\
S &:= \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2 \\
c_1 &:= (m - S^{e_{t_1}} + 1) \bmod t_1 \\
c_2 &:= (m - S^{e_{t_2}} + 1) \bmod t_2 \\
Sig &:= S^{c_1 \cdot c_2} \bmod N
\end{aligned}
$$

$$
\left.
\begin{array}{l}
S = (S \bmod N) + k \cdot N, \\
\text{with } 0 \le k < 2^{161}, \\
\text{if } l(t_1 \cdot t_2) \le 2^{160}
\end{array}
\right\}
$$

$$
\left. Sig = S \bmod N \right\}
$$

- Attack assumes that $k = (S - Sig)/N = t_1 \cdot t_2$
- However, $k < t_1 \cdot t_2$, and any value in $\mathbb{Z}_{t_1 t_2}$ is possible
- Attack does not reveal $t_1$ or $t_2$, it is invalid

SIEMENS

# Wagner's Attack (bit and byte fault version)

- Fault: $m \overset{\lightning}{\mapsto} m + e(m)$,
  bit fault: $e(m) = \pm 2^k$ (byte fault similar)
- Assume that $m$ has been attacked in Line 1:

$$S_p := \tilde{m}^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1$$

- Checking step: $c_1 = (m - \tilde{m}) + 1 \equiv 1 - e(m) \bmod t_1$
- Infective Step: $Sig = S^{c_1}$
- Fact: given $c_1$, we have $p = \gcd(m^{c_1} - Sig^e, N)$
- Fact: for many $e(m)$, reduction mod $t_1$ does not happen
- Fact: all $c_1$ bit faults can be efficiently enumerated
- Fact: scheme can be broken easily

# Enhancement For Bit/Byte Faults

$$
\begin{aligned}
S_p &:= m^{d \bmod \varphi(p \cdot t_1)} \bmod p \cdot t_1 \\
S_q &:= m^{d \bmod \varphi(q \cdot t_2)} \bmod q \cdot t_2 \\
S &:= \mathrm{CRT}(S_p, S_q) \bmod N \cdot t_1 \cdot t_2 \\
c_1 &:= (r_1 \cdot (m - S^{e_{t_1}}) + 1) \bmod t_1 \\
c_2 &:= (r_2 \cdot (m - S^{e_{t_2}}) + 1) \bmod t_2 \\
Sig &:= S^{\gamma} \bmod N
\end{aligned}
$$

} error detection step introducing 2

} $\gamma = c_1 \cdot c_2$

- More general: use $\gamma = f(c_1, c_2, r)$ with $r$ random
- Similar to Ciet, Joye at FDTC 2005: $\gamma = \left\lfloor \frac{r \cdot c_1 + (2^{l(t_i)} - r) \cdot c_2}{2^{l(t_i)}} \right\rfloor$
- Choose $r$ as: $r = d$ or $r = S$ or $r_1 = p, r_2 = q, \ldots$?

SIEMENS

# Conclusion

- BOS-Algorithm facing random faults
  - Attack as presented at CCS'04 contains flaw
  - Scheme remains unbroken
  - Open point: Can we define (and realize) special fault models to break this algorithm?

- BOS-Algorithm facing bit and byte faults
  - Original proposal broken easily
  - We propose several ways to correct weakness (without proof)
  - Still consider such faults as hard to achieve on modern hardware