Attacks
on
Identification and Signature Schemes Involving Corruption of Public Key (Modulus)

Michael Kara-Ivanov, Eran Iceland, Aviad Kipnis

NDS Technologies, Israel Ltd.

FDTC 2008, Washington DC, 10 August

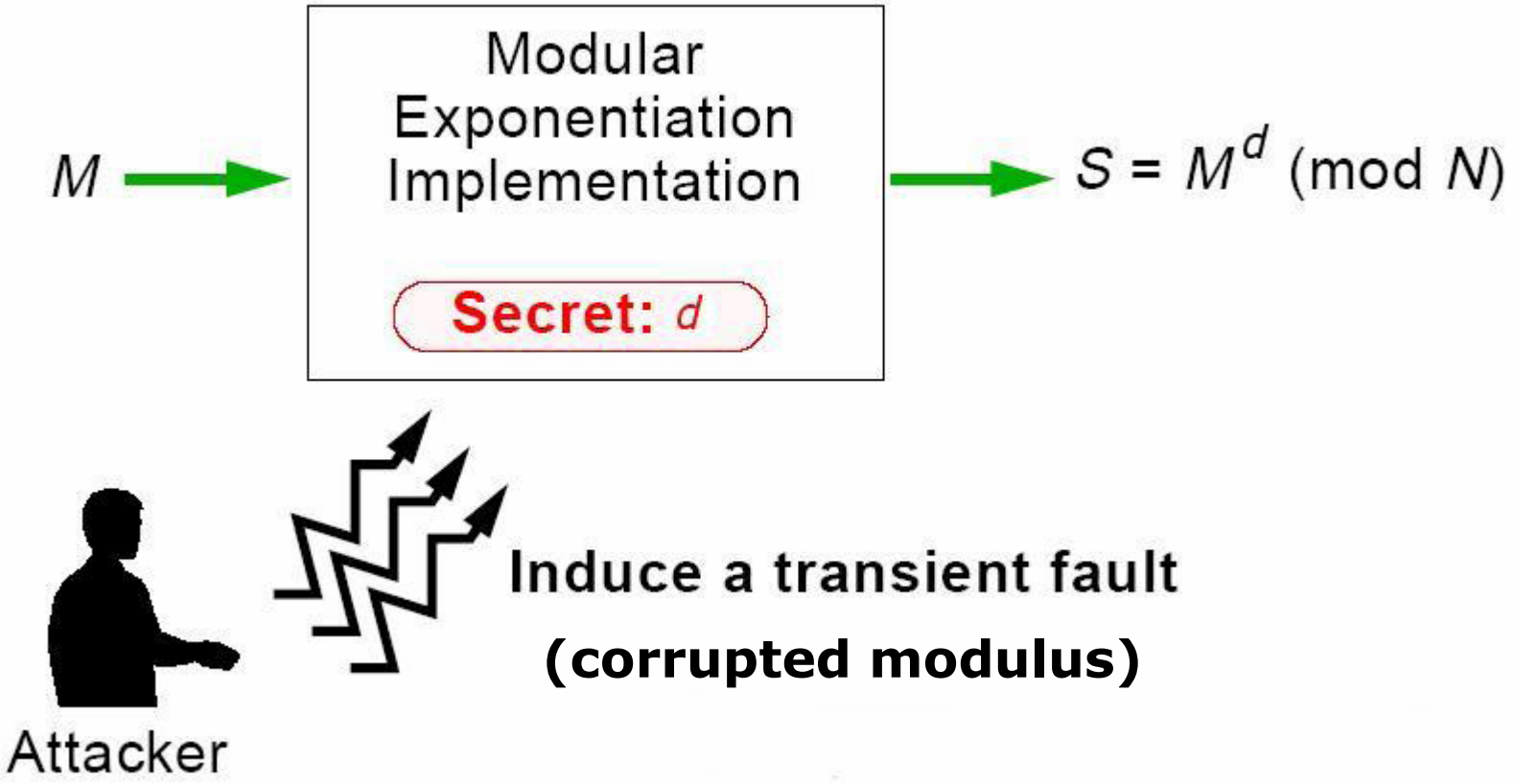
What is it about?

- Fault analysis of public key cryptosystems by corrupting the value of ***public*** parameters (modulus)
- ***We extended*** previous (Brier et al, 2006) attacks to several other signature schemes (ElGamal and OSS) and several identification schemes (Guillou-Quisquater, Schnorr)
- And to ECDSA and DSA (***new idea*** of attack)

Previous Work

- Boneh, DeMillo and Lipton, 1997
- EC: Biehl, Meyer and Muller, 2000
- EC: Blomer, Otto and Seifert, 2005
- RSA: E. Breier, B. Chevallier-Mames, M.Ciet and C. Clavier, Why One Should Also Secure Public Key Elements, (CHES 2006) LNCS 4249, pp 324-338, 2006

Corrupted Modulus Attack on RSA



The Idea:

do (many times)
{

The Attacker: modulus $N \dashrightarrow N_i$ (randomly)

SC evaluates signature $s := m^d \bmod N_i$

}

The Attacker collects the signatures $\{s_i\}_{i=1\ldots}$

for fixed m and (unknown) random $\{N_i\}_{i=1\ldots}$

The fact:

For any little prime p the statistics of $\{s_i\}$ supplies

$$d_p = d \bmod (p-1)$$

CRT:

When the Attacker knows a d_p for enough primes p such that $LCD(p-1) > d$, he can evaluate d.

Explanation of the idea

For

- fixed small prime q
- uniformly distributed big N_i such that $q << N_i$

Let's examine the values

$$V_N := (m^d \bmod N_i) \bmod q$$

What is their distribution?

- For N_i s.t. $q | N_i$ $V_N = m^d \bmod q$ and does not depend on N_i
- Otherwise V_N is distributed ~ uniformly

Therefore…

Explanation of the idea - Cont

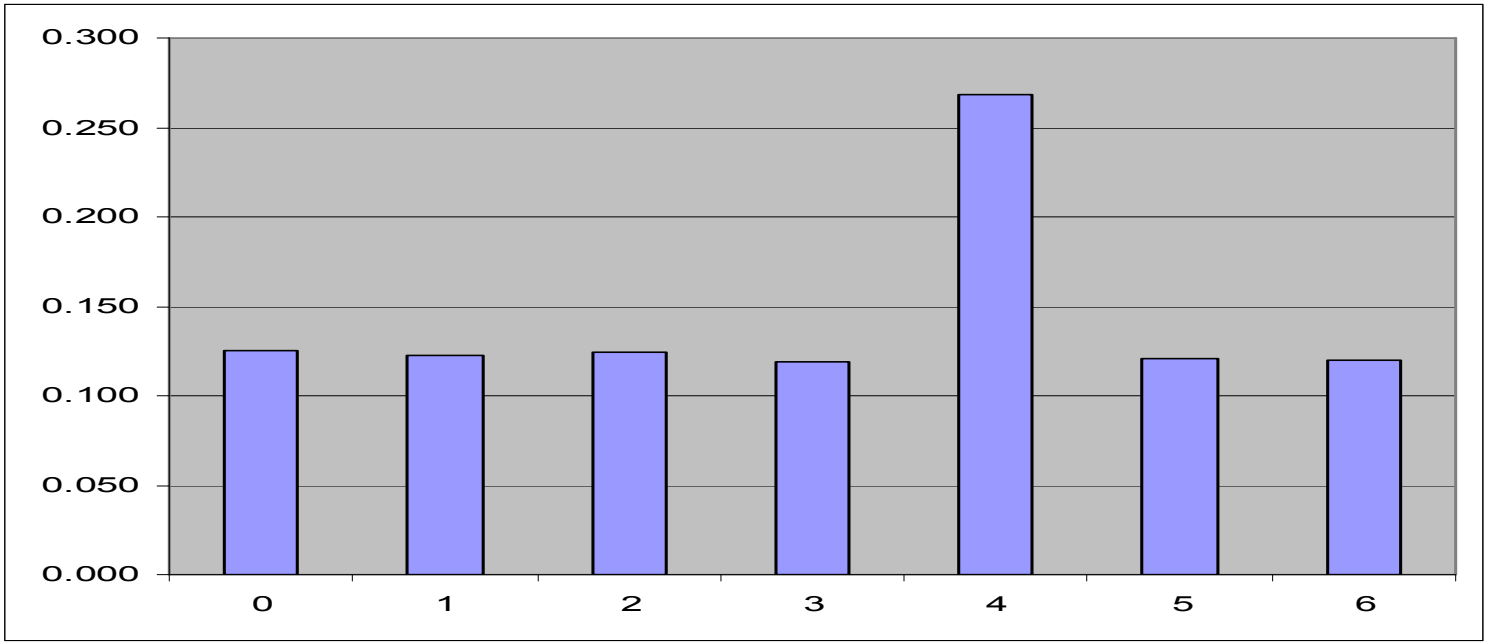
For any $r=0,1,\ldots,q-1$

$Prob\{V_N=r\} \approx$

$(2q-1)/q^2$ - if $r = m^d \bmod q$

$(q-1)/q^2$ - if $r \neq m^d \bmod q$

Typical histogram ($q=7$, $m^d = 2^{32}$, $N=1000\ldots25000$):



The Fault Model

- The public key is stored in a non-volatile memory of a cryptographic device (e. g., in the EEPORM of a smart card).

- **When Modulus N** is loaded: EEPROM -> RAM, a random fault is injected **N** -> $\mathbf{N_i}$.

- $\mathbf{N_i}$ is used by the device (instead of **N**) to produce a corrupted signature or corrupted intermediate identification parameters.

- Attacker collects many corrupted signatures or intermediate parameters of the identification protocols: recovers the secret key.

- Note: This discussion relates to corruption of the public key parameters (which are usually less defended); The attacker does not need to know the value of the corrupted modulus.

Guillou-Quisquater Identification Scheme

Key Generation

Assume that Peggy wants to prove her identity to Victor.

Peggy's public key :

J - a set of credentials,

N - product of two big primes

v - a large exponent s. t. $\gcd(v, \varphi(N)) = 1$

Peggy's private key:

B, calculated such that $JB^v = 1 \bmod N$.

Identification Scheme

- Peggy picks a random integer r in the interval $[1, N-1]$;
- Peggy computes $T := r^v \bmod N$ and sends it to Victor.
- Victor picks a random integer d in the interval $[0, v-1]$. Victor sends d to Peggy.
- Peggy computes $D := rB^d \bmod N$ and sends it to Victor.
- Victor computes $T' := D^v * J^d \bmod N$.

If $T = T' \bmod N$, then identification succeeds.

Our Attack on GQ – Extension of Brier's (2006) Idea

do (many times)
{

The Attacker: $\mathbf{N} \dashrightarrow \mathbf{N}_i$ (corrupted: assume uniform distribution)

Peggy : $\mathbf{T}_i$ to Victor.

Victor: $\mathbf{d}_i$ to Peggy.

Peggy : $\mathbf{D}_i$ to Victor.

The Attacker collects the triples $\{\mathbf{T}_i, \mathbf{D}_i, \mathbf{d}_i\}$

}

For each little prime $\mathbf{q}$ s. t. $\mathbf{gcd(q-1, v)=1}$ - the Attacker :

evaluates $\mathbf{B}_{iq} := \mathbf{D}_i^{1/d_i} * \mathbf{T}_i^{-1/(vd_i)} \pmod{q}$

and examines the distribution of $\{\mathbf{B}_{iq}\}$:

The fact:

$\mathbf{B}_{iq}$ statistics supplies $\mathbf{B} \pmod{q}$

CRT:

The Attacker recovers $\mathbf{B}$ from $\{\mathbf{B} \pmod{q}\}$ for enough $\mathbf{q}$'s.

An Attack on Guillou-Quisquater Identification Scheme - Cont.

- For each triple, if $\mathbf{gcd(d_i, q-1)=1}$, evaluate

$$\mathbf{B_i := D_i^{1/d_i} * T_i^{-1/(vd_i)} \bmod q}$$

- The repetition of the results corresponds to a case when $\mathbf{q \mid N_i}$, and thus the Attacker knows $\mathbf{B \pmod q}$
- Otherwise $\mathbf{B_i \pmod q}$ are uniformly distributed
- By CRT: recover $\mathbf{B}$ from partial residues $\mathbf{B \pmod q}$

ECDSA keys

The Curve parameters: (p,a,b,G,N,h) :

- p - prime number.
- The curve' s equation: $y^2=x^3+ax+b$ over $E(F_p)$.
- $G = (x_G, y_G)$ - base point on $E(F_p)$.
- Prime N - the order of G.
- $h = \#E(F_p)/N$ - the cofactor.

The secret key:

d – integer number in the range [1, N-1].

The public key:

Q – point on the curve s. t. $Q=d*G$

ECDSA Protocol

To sign a message **m** with the hash **e**:

I. Select a random integer **k** in the range **[1,N-1]**.

II. Evaluate **$r := x_1 \bmod N$**, where **$(x_1, y_1) = k*G$**.

III. If **r = 0**, go back to step I.

IV. Evaluate **$s := k^{-1}(e + r*d) \bmod N$**.

V. If **s=0**, go back to step I.

The signature is the pair **(r,s)**.

Simple Attack on Modulus Did NOT Succeed

Attacks in the Previous Work

- Previous work attacked the secret key mostly when calculating **`Q = d*G`**.

- For example:

  Biehl, Meyer and Muller, 2000

  Blomer, Otto and Seifert, 2005

  Ciet and Joye, 2005

- We propose revealing the secret key by attacking the public elements (modulus) when generating signatures.

Our Attack on ECDSA – **a New Idea** – "make use" of "$s:=k^{-1}(e + r*d) \bmod N$"

do (many times)

{

The Attacker: modulus $N \dashrightarrow N_i$ (randomly)

SC evaluates signature with corrupted N_i

The Attacker collects the corrupted signatures (r_i, s_i)

}

The Attacker, for given small prime q:

- leaves only signatures for which $s_i=0 \bmod q$ and $r_i \neq 0 \bmod q$
- evaluates $d_i:=-e/r_i \pmod q$ as a candidate for $d \bmod q$

The fact:

Corrupted signatures statistics supplies $d \pmod q$

CRT:

The Attacker recovers unreduced value of d from partial residues $d \pmod q$

Our Attack on ECDSA –Cont.

Let q be a small prime and suppose that:

- $s_i = 0 \mod q$
- $r_i \neq 0 \mod q$

Let $d_i := -e/r_i \pmod q$ (Note that $d_i \neq 0$)

From step IV : $k^{-1}(e + r_i d) = 0 \pmod q$

Thus with significant probability $e + r_i d = 0 \pmod q$ and thus $d = d_i \pmod q$

Let us examine the $\{d_i\}$ distribution. There are 2 possibilities:

a) If $d = x \mod q$, for $x \neq 0$, the distribution has a distinct peak (almost twice more) at the value x.

b) If $q \mid d$, the distribution is approximately uniform (except value 0).

By CRT – recover d from $\{d \mod q\}_q$.

The Amount of Data that Should Be Collected – SW Simulation Results

	Secret Key Length (bits)	Faulty Signatures	Max Prime (for recovery)
RSA Brier, 2006	1024	60K	3593
ECDSA	160	250K	131
ECDSA	256	600K	199
DSA	160	250K	131
GQ - Ident	1024	60K	743
ElGamal - Sig	1024	~ 15 M	843
Schnorr - Ident	160	~ 524 M	389

Possible Protection Measures

- Do not expose faulty values (e.g., signatures).
- Evaluate twice or check the consistency before exposing data: partial verification (like checksum) may be enough.
- Identification schemes: encrypt communication.
- Hash intermediate values.
- Transfer the modulus NVM -> RAM several times in various identification and signature stages.