Fault Attacks on Signature Schemes

Christophe Giraud
Oberthur Card Systems

Erik Knudsen
Logos Smart Card A/S

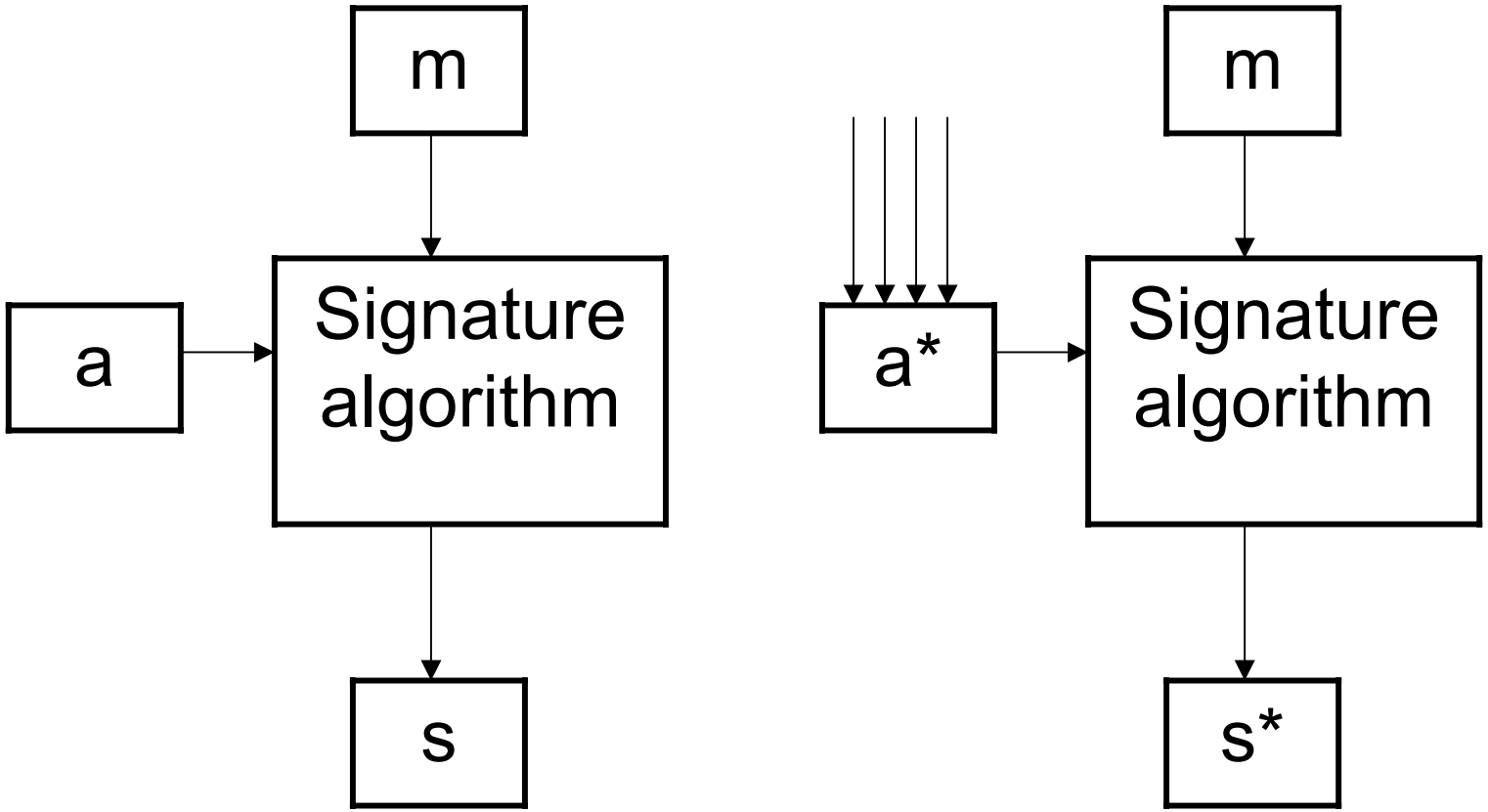
Contribution

- Extension of the bit-fault model of Bao et. al.
- Attacks on DSA, ElGamal and Schnorr (in this presentation only DSA)
- Upper bound for number of faulty signatures

Byte fault model

- Weaker assumption than bit-fault
- Easier to put into practice
- Bit-faults have been reported but better chips are being manufactured

Fault attacks



Strategy

- Induce a fault on one byte of the key
- Find the difference between the correct and the faulty byte in the key
- Restrict the number of possible values for each byte of the key to at most n
- Retrieve the key by exhaustive search among n^m candidates (m = #bytes in key)

DSA-parameters

- Public parameters: (p, q, g)

- Primes: q 160-bit, p 1024-bit
- Base point: $1 < g \leq p-1$, $g^q \equiv 1 \pmod{p}$

- Private key: $1 \leq a \leq q-1$ (20 bytes)
- Public key: $A = g^a \bmod p$

DSA-signature

- hash of message: h
- random: $1 \leq k \leq q-1$

- $r = (g^k \bmod p) \bmod q$
- $s = k^{-1}(h+ar) \bmod q$

- Signature of message: (r,s)

DSA-verification

- hash of message: h

- $b = A^{r/s} \bmod p = g^{ar/s} \bmod p$
- $c = g^{h/s} \bmod p$
- $T = bc \bmod p \bmod q$

- Accept iff T=r

Erroneous key

- $a^* = a + e_{i,j}$
- $e_{i,j} = j.2^{8i}$, $0 \leq i \leq 19$, $-255 \leq j \leq 255$
- (r, s^*), $s^* = k^{-1}(h + a^* r) \bmod q$
- First task: determine $e_{i,j}$

A useful value

- $b = A^{r/s^*} = g^{ar/s^*}$, $c = g^{h/s^*}$
- $T = bc \bmod p \bmod q$ (DSA-verification)

- $T = g^{(h+ar)/s^*} \bmod p \bmod q$
- $R_{i,j} = g^{e_{i,j}.r/s^*} \bmod p \bmod q$

- $TR_{i,j} = g^{(h+r(a+e_{i,j}))/s^*} \bmod p \bmod q$
  $= g^{k(h+r(a+e_{i,j})) / (h+ra^*)} \bmod p \bmod q$

How do we pin-point the error?

- $TR_{i,j} = g^{k(h+r(a+e_{i,j})) / (h+ra^*)} \bmod p \bmod q$
  $= g^{kf_{i,j}} \bmod p \bmod q$

- $f_{i,j} = (h+r(a+e_{i,j})) / (h+ra^*)$
- $f_{i,j} = 1$ iff $a+e_{i,j} = a^*$

- $a^* = a+e_{i,j}$ iff $TR_{i,j} = r \bmod p \bmod q$

The faulty byte

- i’th byte of the key: $e = a^*_i - a_i$

- bytes: $0 \leq a^*_i, a_i \leq 255$
- range for e: $-a_i \leq e \leq 255 - a_i$

- range for a_i: $-e \leq a_i \leq 255 - e$

A sufficient condition

256-n ≤ e(s)-e(t)

restricts a_i to at most n values:

- $\forall e$: $-e \leq a_i \leq 255-e$
- $\underline{-e(t)} \leq a_i \leq 255-e(s) \leq \underline{(n-1)- e(t)}$

How many faults do we need?

Define T_n as:

$$\min\{ t \geq 2 \mid \exists\ 1 \leq s < t : |e(t)-e(s)| \geq 256-n\}$$

Two assumptions:

1. Faults are mutually independent
2. $\forall$ e: Prob(e) = 1 / 256

Theoretical Result

- $p = 1 / 256$
- $P(T_n \leq t) = 1-(n+1)(1-np)^t+n(1-(n+1)p)^t$
- $E[T_n] = (2n+1) / (np(n+1))$

Upper bound

Av. number of faulty signatures	App. number of keys left
7680	1
4267	2^{20}
2987	2^{32}
2304	2^{40}

Idea of proof

for $1 \leq j \leq n$ define events C_j :

“with t faulty signatures we have observed
$e = -a_i+j-1$ and $255-a_i-n+j \leq e' \leq 255-a_i$”

$\{T_n \leq t\} = C_1 U \ldots U C_n$

Inclusion-Exclusion Principle

$$P(T_n \leq t) = P(C_1 U \ldots U C_n)$$

$$= \sum (-1)^{k+1} \sum P(\cap C_i)$$

- and lots of calculations…

Conclusion

- Byte fault models are feasible
- At most 2304 faulty signatures to break DSA