Known-Plaintext-Only Attack on RSA-CRT with Montgomery Multiplication

Martin Hlaváč

Department of Algebra, Charles University in Prague
IT Security Department, Czech Insurance Corporation

September 7, CHES 09, Lausanne, Switzerland

Outline

- Electronic Passport
- Electro-Magnetic Interface
- Active Authentication
- RSA-CRT with Mont. Multiplication
- Schindler's Attack, Tomoeda's Attack
- New attack
- Simulation Results
- Conclusion

Outline

- Electronic Passport
- Electro-Magnetic Interface
- Active Authentication
- RSA-CRT with Mont. Multiplication
- Schindler's Attack, Tomoeda's Attack
- New attack
- Simulation Results
- Conclusion

Motivation
&
strong
assumptions

Outline

- Electronic Passport
- Electro-Magnetic Interface
- Active Authentication
- RSA-CRT with Mont. Multiplication
- Schindler's Attack, Tomoeda's Attack
- New attack
- Simulation Results
- Conclusion

Motivation
&
strong
assumptions

Attack

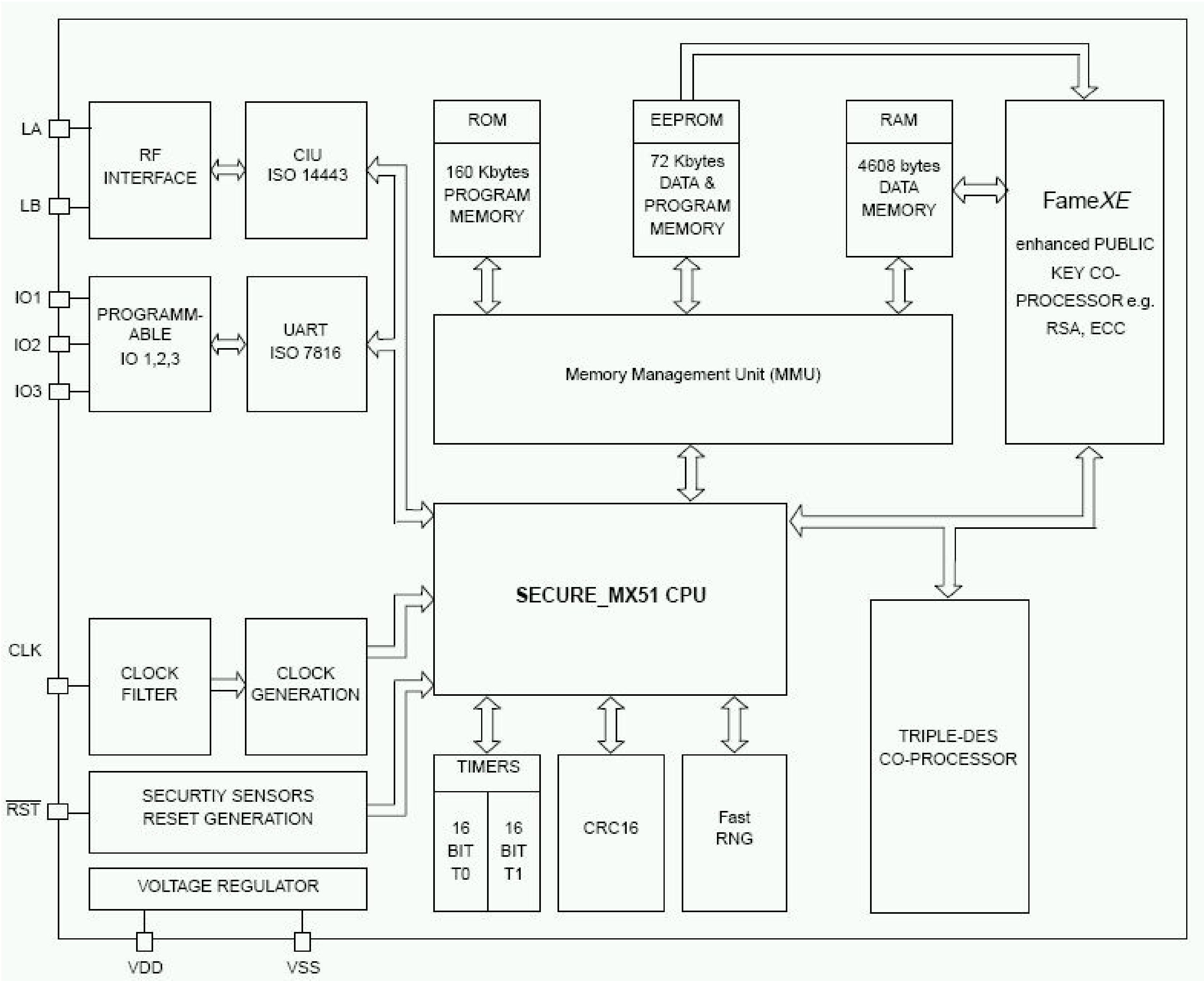
Post-Conference Remarks!!

- Based on the discussions with Jean-Jacques Quisquater and an NXP representative during and after CHES 2009 in Lausanne regarding the assumptions made in the presentation, the author completes these slides with the remarks shown in red
- Specifically, Jean-Jacques Quisquater states
  - FameXE cryptographic coprocessor should never use Montgomery multiplication
  - If someone implemented Montgomerry multiplication on top of FameXE, it was highly unusual and unfortunate decision
  - Fame-X uses “Quisquater” multiplication algorithm
- NXP confirms
  - **Electronic passport referred to in this work does NOT use Montgomery multiplication algorithm**
- As a result, these facts do not affect the validity of the mathematical attack described here. It can not be applied in the electronic passport scenario, however.

Electronic Passport

- Electronic travel document with chip that contains
  - MRZ (Machine Readable Zone)
  - Photo, fingerprints?, eye retina?
  - RSA Key pair for Active Authentication

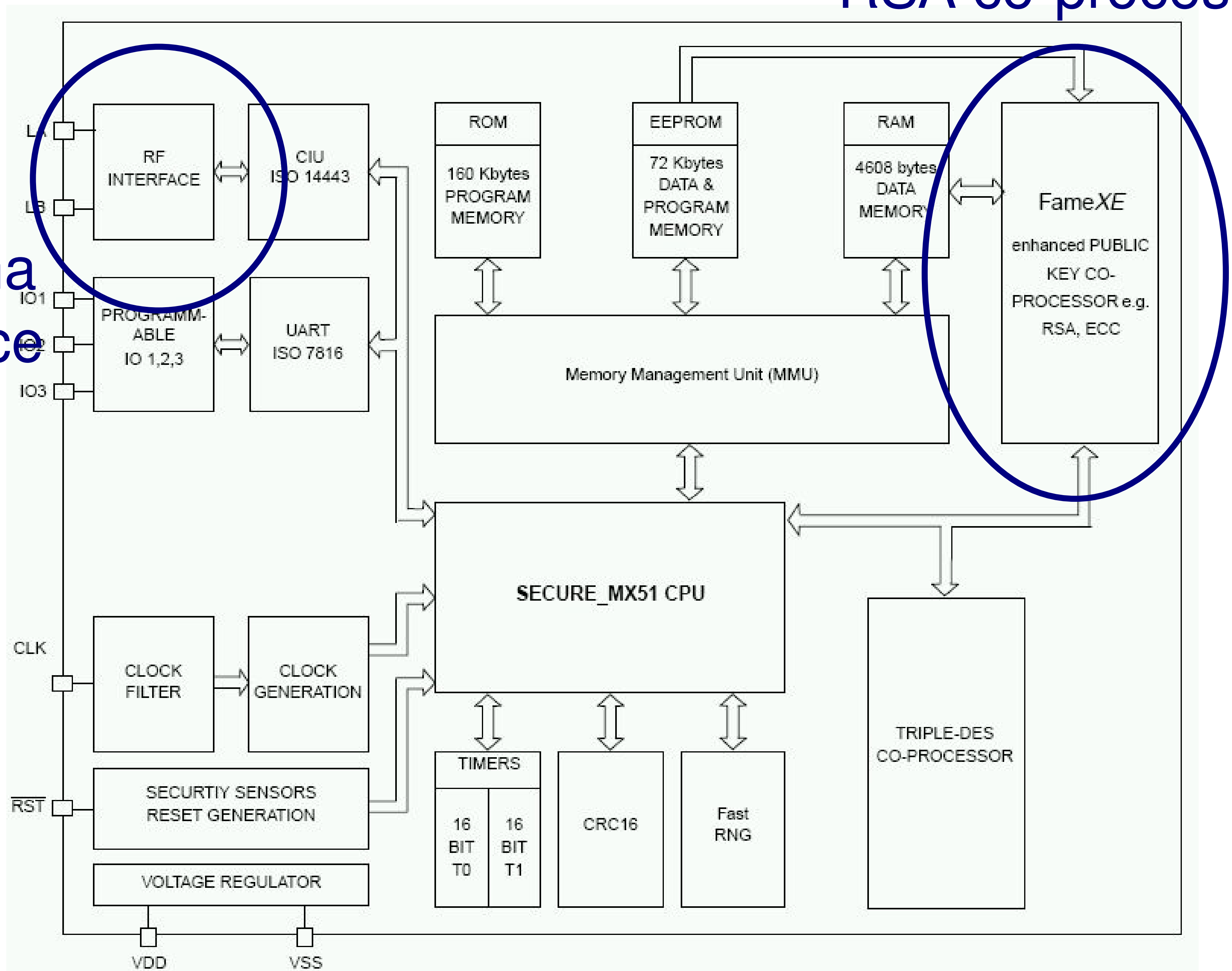
Electronic Passport II.



Electronic Passport II.

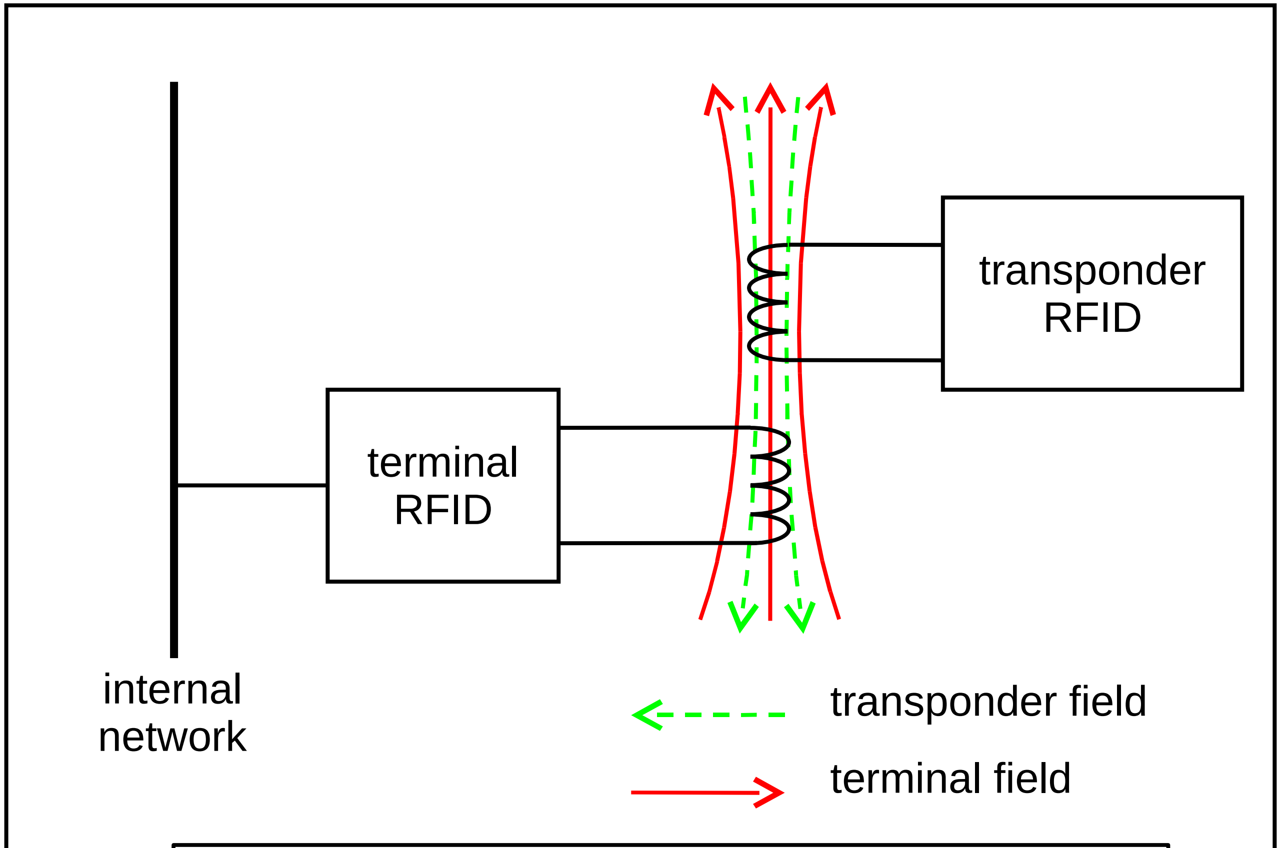
FameXE
RSA co-processor

antenna
interface



Electro-Magnetic Interface

- HF range (13.56 MHz)
- Operation distance ~10 cm (4 in)
- Near-field communication



Can be seen as “high frequency transformer”

Active Authentication

- passport's anti-cloning countermeasure
- optional
- simple challenge-response protocol based on RSA
  - public key signed by national authority
  - private key in protected memory
- challenge chosen by **both**, passport and reader
- chosen plaintext attacks do not work

Active Authentication II.

Algorithm 1 Active authentication

Parties: $\mathbf{T}$ ... terminal, $\mathbf{P}$ ... passport

1: $\mathbf{T}$**:** generate random 8-byte value V
2: $\mathbf{T} \rightarrow \mathbf{P}$**:** V
3: $\mathbf{P}$**:** generate random 106-byte value U
4: $\mathbf{P}$**:** compute $s = m^d \ \mathbf{mod}\ N$, where $m =$"6A"$||U||w||$"BC", $w = \texttt{SHA-1}(U||V)$ and d is the passport's secret AA key securely stored in the protected memory
5: $\mathbf{P} \rightarrow \mathbf{T}$**:** s, U
6: $\mathbf{T}$**:** verify $m = s^e \ \mathbf{mod}\ N$, where e is the passport's public key stored in publicly accessible part of passport memory

Active Authentication II.

Algorithm 1 Active authentication

Parties: **T** . . . terminal, **P** . . . passport

1: **T:** generate random 8-byte value V

2: **T** $\rightarrow$ **P:** V

3: **P:** generate random 106-byte value U

4: **P:** compute $s = m^d \bmod N$, where $m =$ "6A"$||U||w||$"BC", $w =$ SHA-1$(U||V)$ and d is the passport's secret AA key securely stored in the protected memory

5: **P** $\rightarrow$ **T:** s, U

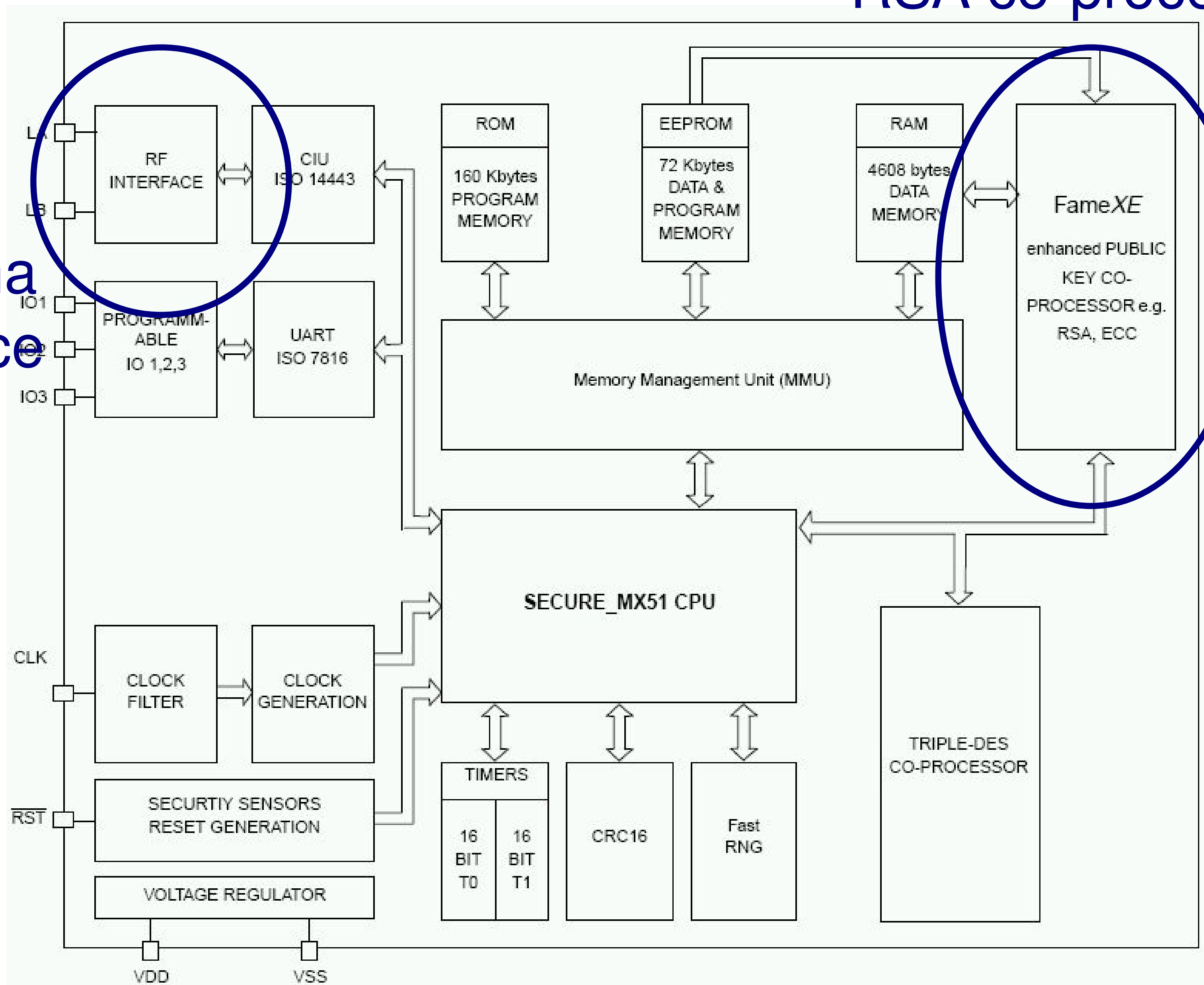
6: **T:** verify $m = s^e \bmod N$, where e is the passport's public key stored in publicly accessible part of passport memory

chosen jointly

Electronic Passport II.

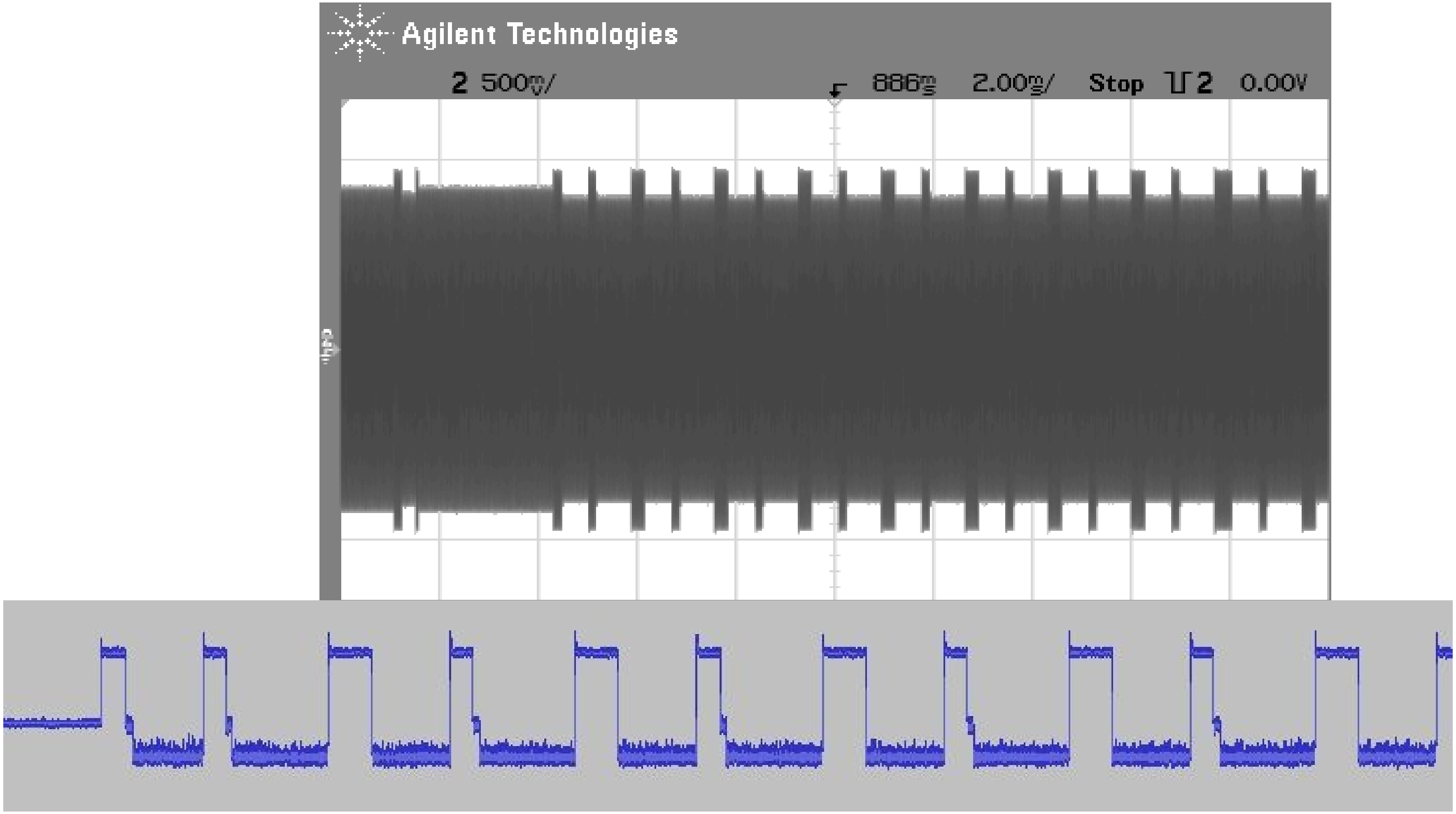
FameXE
RSA co-processor

antenna
interface



FameXP exposure in EM field

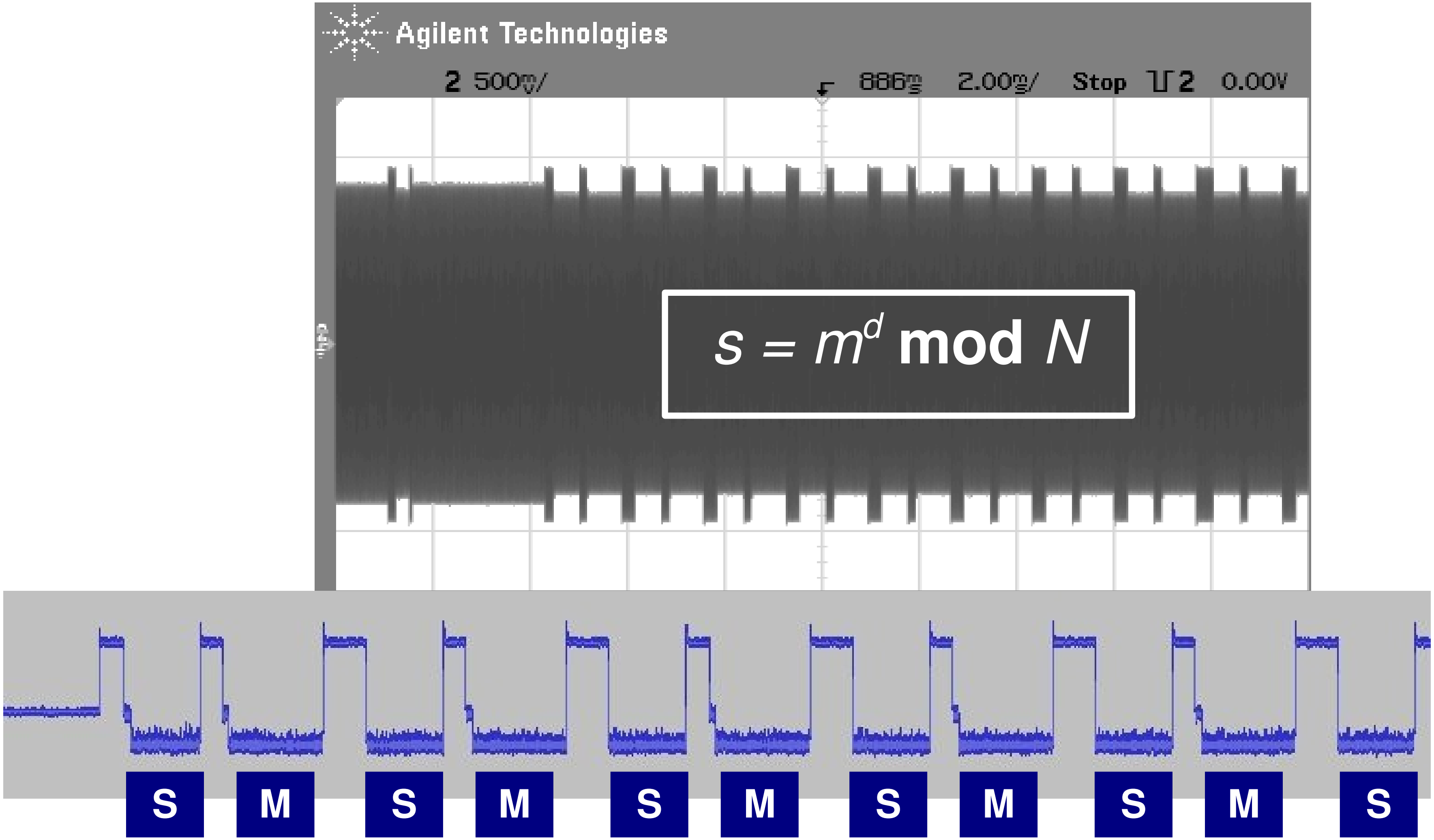
- Sensitive operation should **not** be visible



Measurements by doc. Lórencz's team, FEL CTU in Prague, april 2007

FameXP exposure in EM field

- Sensitive operation should **not** be visible



Measurements by doc. Lórencz's team, FEL CTU in Prague, april 2007

RSA with Chinese Remainder Theorem

- Private RSA operation m^d **mod** N is computed using CRT as follows

$$s_p = (m_p)^{d_p} \mathbf{mod}\, p$$
$$s_q = (m_q)^{d_q} \mathbf{mod}\, q$$
$$s = ((s_q - s_p)\, p_{inv} \,\mathbf{mod}\, q)\, p + s_p$$

- Faster then simple exponentiation
- Use of secret *p*,*q* make CRT vulnerable

RSA with Chinese Remainder Theorem

- Private RSA operation m^d **mod** N is computed using CRT as follows

$$s_p = (m_p)^{d_p} \bmod p$$
$$s_q = (m_q)^{d_q} \bmod q$$
$$s = ((s_q - s_p)\, p_{inv} \bmod q)\, p + s_p$$

- Faster then simple exponentiation
- Use of secret *p*,*q* make CRT vulnerable

Assumption n. 1:
RSA blinding is NOT employed.
(Time consuming.)

Montgomerry exponentiation

Algorithm 3 Montgomery exponentiation *expmont()*

Input: $m, p, d\left(= (d_{n-1}e_{d-2}\dots d_1 d_0)_2\right)$
Output: $x = m^d \bmod p$

$u \leftarrow mR \bmod p$
$z \leftarrow u$
for $i \leftarrow n-2$ **to** 0
  $z \leftarrow mont(z, z, p)$
  if $d_i == 1$ **then**
    $z \leftarrow mont(z, u, p)$
  else
    $z' \leftarrow mont(z, u, p)$
endfor
$z \leftarrow mont(z, 1, p)$
return z

- mod&div R=2^{512}
- fast

Algorithm 2 Montgomery multiplication *mont()*

Input: $x, y \in Z_p$
Output: $w = xyR^{-1} \bmod p$

$s \leftarrow xy$
$t \leftarrow s(-p^{-1}) \bmod R$
$g \leftarrow s + tp$
$w \leftarrow g/R$
if $w > p$ **then**
  $w \leftarrow w - p$ (final subtraction)
return w

Montgomerry exponentiation

Algorithm 3 Montgomery exponentiation *expmont()*

Input: $m, p, d\ (= (d_{n-1} e_{d-2} \ldots d_1 d_0)_2)$
Output: $x = m^d \bmod p$

```
u ← mR mod p
z ← u
for i ← n − 2 to 0
        z ← mont(z, z, p)
        if d_i == 1 then
                z ← mont(z, u, p)
        else
                z' ← mont(z, u, p)
endfor
z ← mont(z, 1, p)
return z
```

- mod&div R=2^{512}
- fast

Algorithm 2 Montgomery multiplication *mont()*

Input: $x, y \in Z_p$
Output: $w = xyR^{-1} \bmod p$

```
s ← xy
t ← s(−p^{-1}) mod R
g ← s + tp
w ← g/R
if w > p then
        w ← w − p (final subtraction)
return w
```

Assumption n. 2:
No constant-time
multiplication is used.

Montgomerry exponentiation

Algorithm 3 Montgomery exponentiation *expmont()*

Input: $m, p, d\ (= (d_{n-1}e_{d-2} \ldots d_1 d_0)_2)$
Output: $x = m^d \bmod p$

```
1:  u ← mR mod p
2:  z ← u
3:  for i ← n − 2 to 0
4:        z ← mont(z, z, p)
5:        if d_i == 1 then
6:              z ← mont(z, u, p)
7:        else
8:              z' ← mont(z, u, p)
9:  endfor
10: z ← mont(z, 1, p)
11: return z
```

Assumption n. 2: No constant-time multiplication is used.

- mod&div R=2^{512}
- fast

See remarks on slide 5.

Algorithm 2 Montgomery multiplication *mont()*

Input: $x, y \in Z_p$
Output: $w = xyR^{-1} \bmod p$

```
1: s ← xy
2: t ← s(−p^{−1}) mod R
3: g ← s + tp
4: w ← g/R
5: if w > p then
6:       w ← w − p (final subtraction)
7: return w
```

Schindler's Timing Attack

- With x fixed and B random in $\boldsymbol{Z}_p$, the probability final subtraction occurs during mont(x,B) is

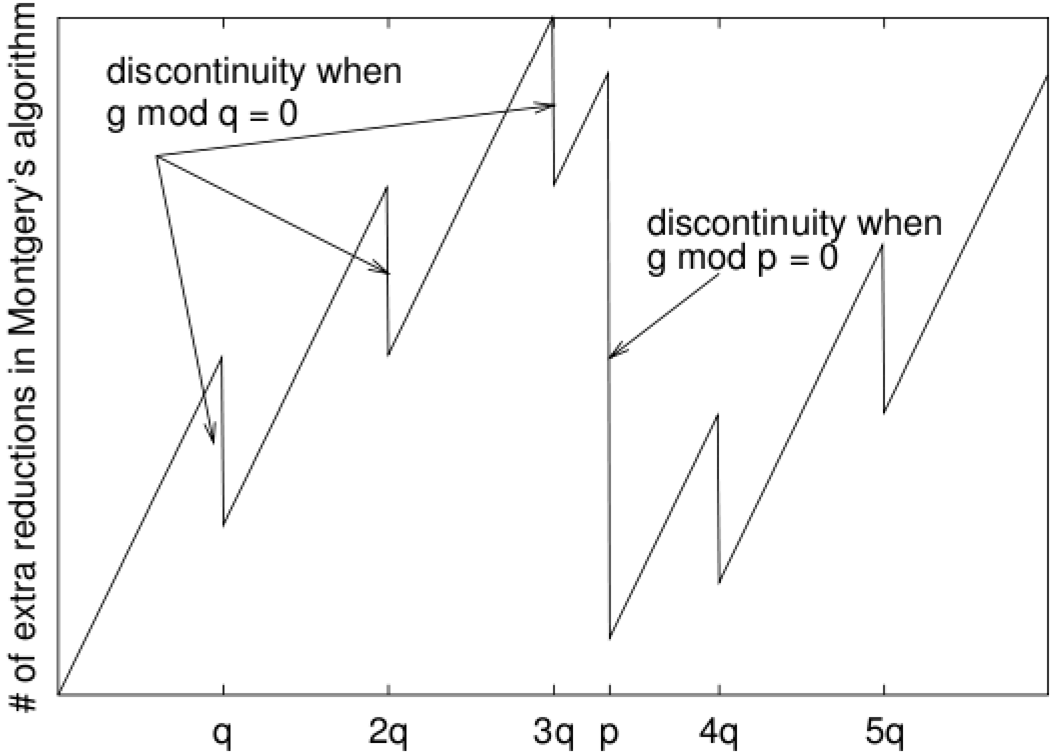
$$\frac{x \bmod p}{2R}$$

Schindler's Timing Attack

- With *x* fixed and *B* random in $\mathbf{Z}_p$, the probability final subtraction occurs during mont(*x*,*B*) is

$$\frac{x \bmod p}{2R}$$

- Careful choice of plaintext allows **timing** ACPA.



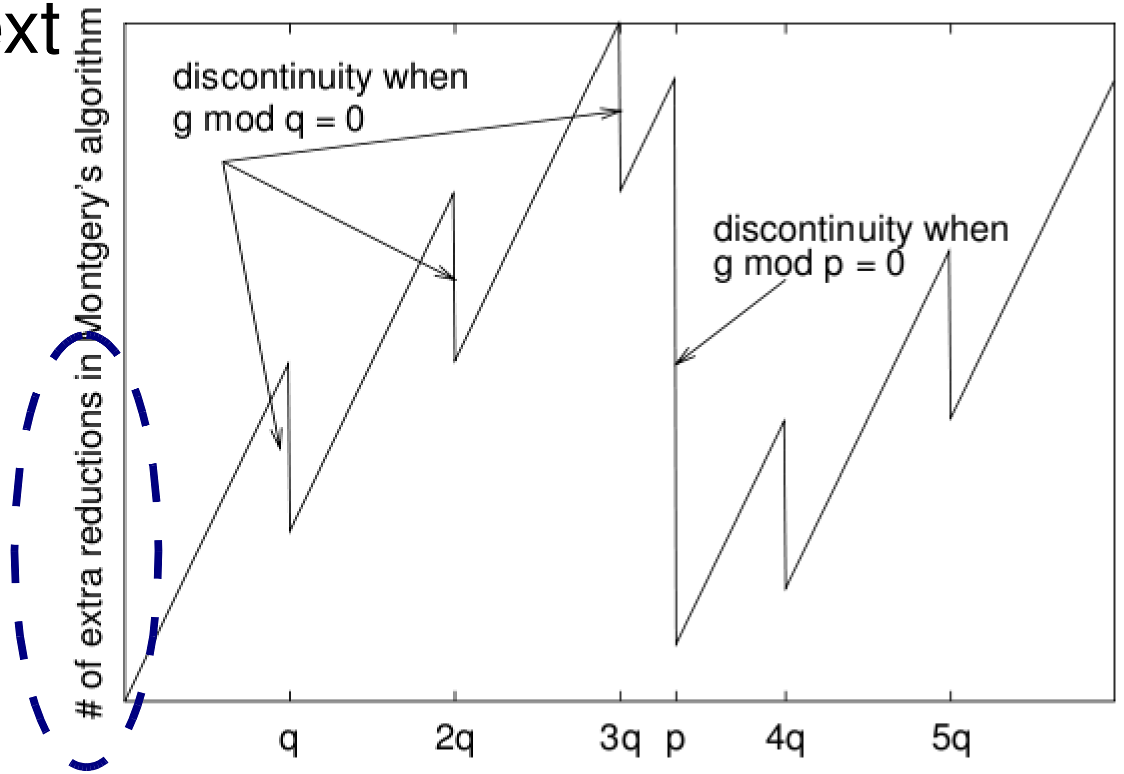
Schindler's Timing Attack

- With x fixed and B random in $\boldsymbol{Z}_p$, the probability final subtraction occurs during mont(x,B) is

$$\frac{x \bmod p}{2R}$$

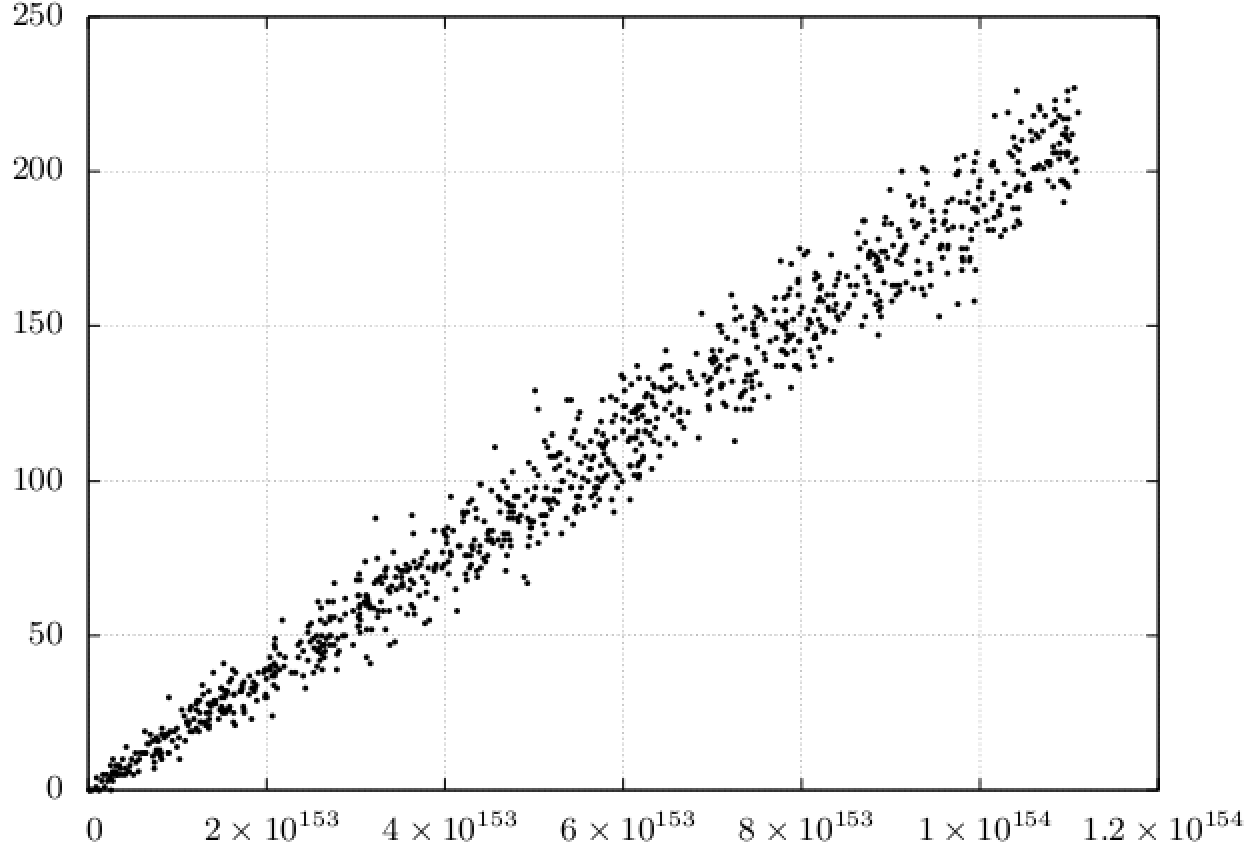
- Careful choice of plaintext allows **timing** ACPA.

Assumption n. 3:
Amount of final
subtractions is revealed.

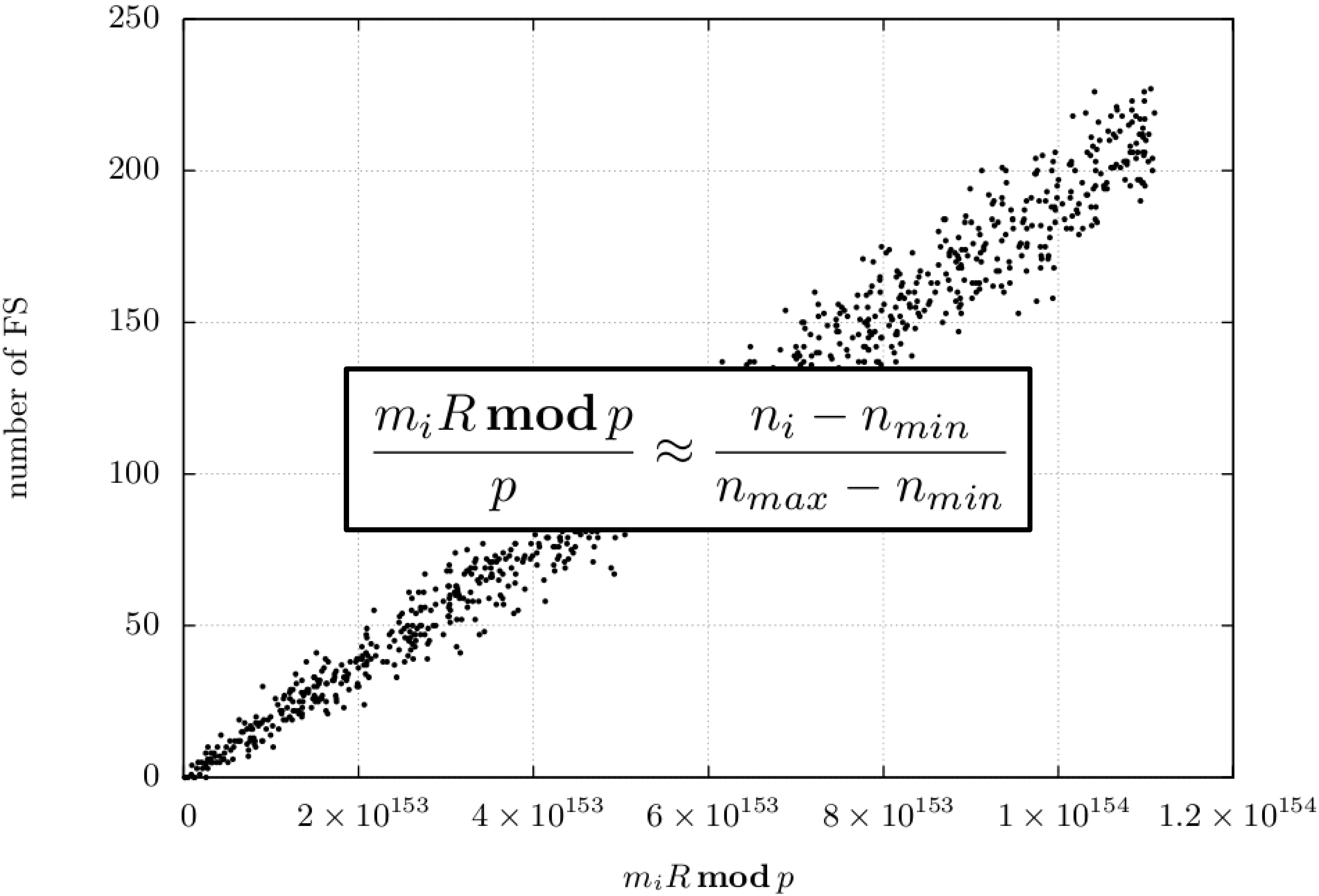


CPA attack by Tomoeda et al.

- Tomoeda et al. observed "almost linear" modular relation between
  - (function of) **unknown** factor *p*, and
  - **known** amount of final substitutions



CPA attack by Tomoeda et al. II.



$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

New attack – Basic Idea

$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

New attack – Basic Idea

$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

- Multiply by *N*

$$m_i R q - k_i N \approx \frac{n_i - n_{min}}{n_{max} - n_{min}} N$$

New attack – Basic Idea

$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

- Multiply by *N*

$$m_i R q - k_i N \approx \frac{n_i - n_{min}}{n_{max} - n_{min}} N$$

- Substitute

$$t_i = m_i R \,\mathbf{mod}\, N$$
$$u_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} N$$

$$|t_i q - u_i|_N \le \frac{N}{2} 2^{-l_i}$$

New attack – Basic Idea

$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

- Multiply by *N*

$$m_i R q - k_i N \approx \frac{n_i - n_{min}}{n_{max} - n_{min}} N$$

- Substitute

$$t_i = m_i R \,\mathbf{mod}\, N$$
$$u_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} N$$

$$|t_i q - u_i|_N \leq \frac{N}{2} 2^{-l_i}$$

- use Hidden Number Problem

$$\mathbf{B} = \begin{pmatrix} N & 0 & \cdots & 0 & 0 \\ 0 & N & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & N & 0 \\ t_0 & \cdots & \cdots & t_{k-1} & N^{\frac{1}{2}}/2^{l+1} \end{pmatrix}$$

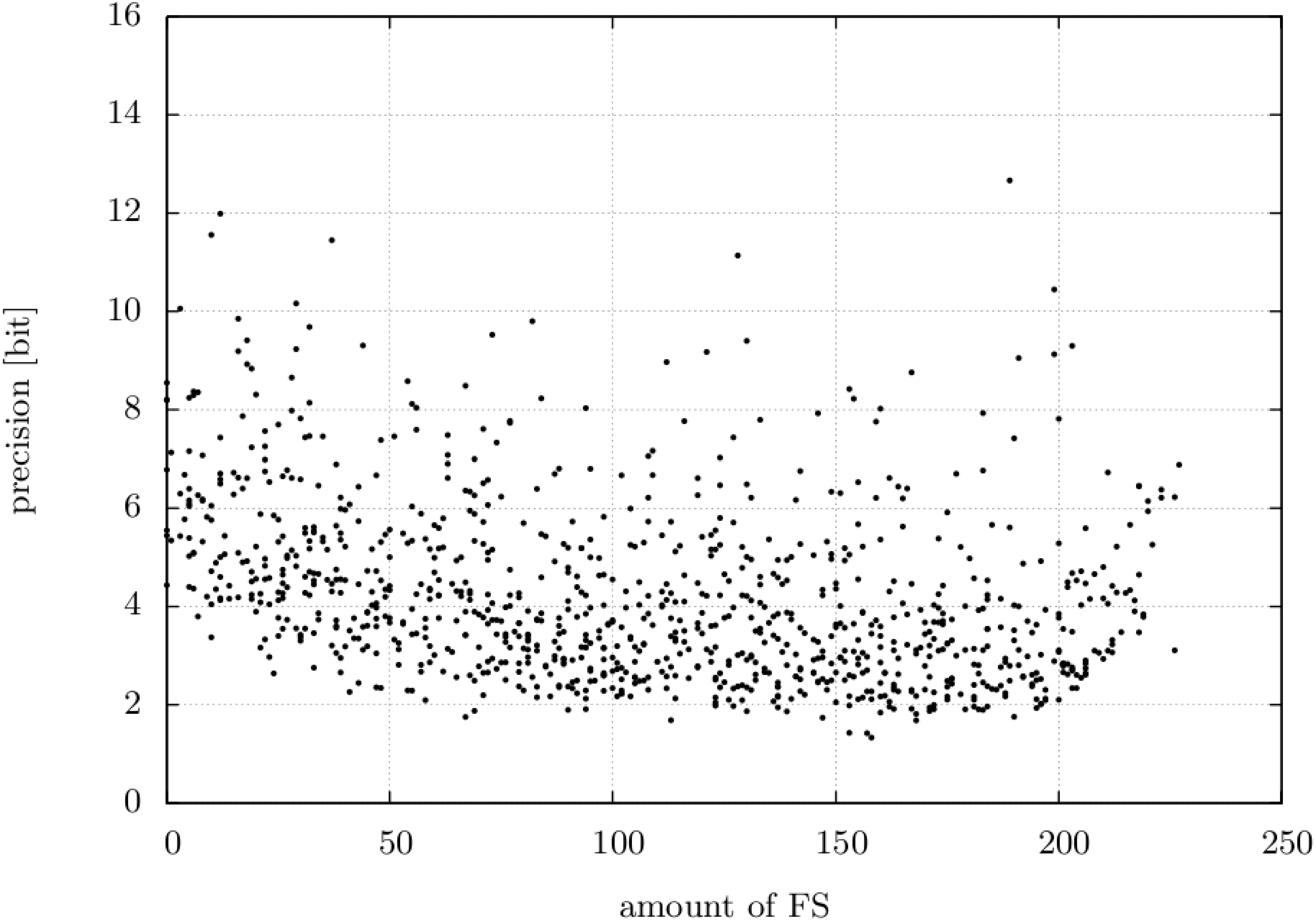
Precision of approximations

- How good is "≈" in $\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$?

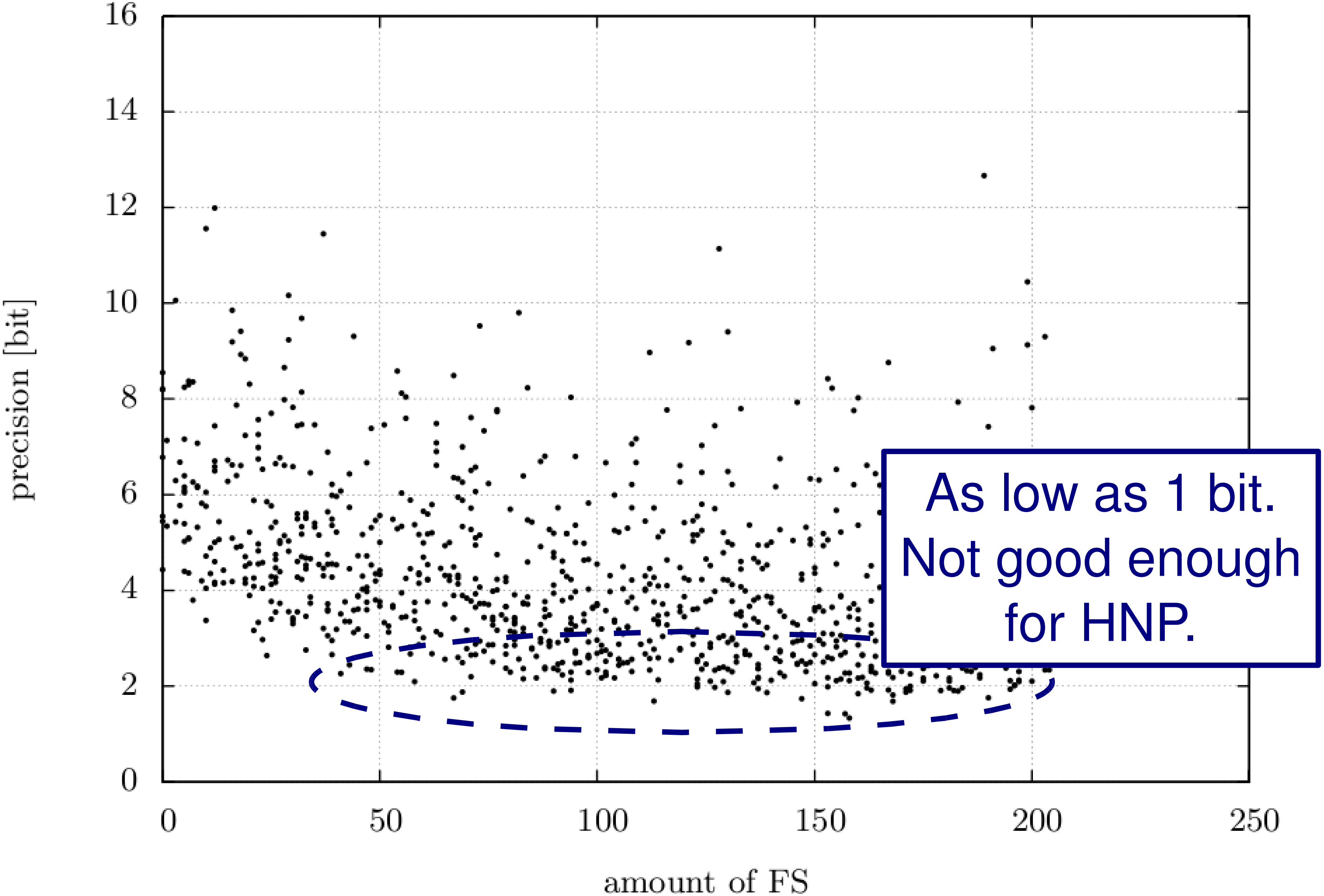
- one-time pre-computation with
  - 2^{12} RSA instances
  - 2^{12} signatures per instance

- *n* bit precision if difference below 2^{-n}

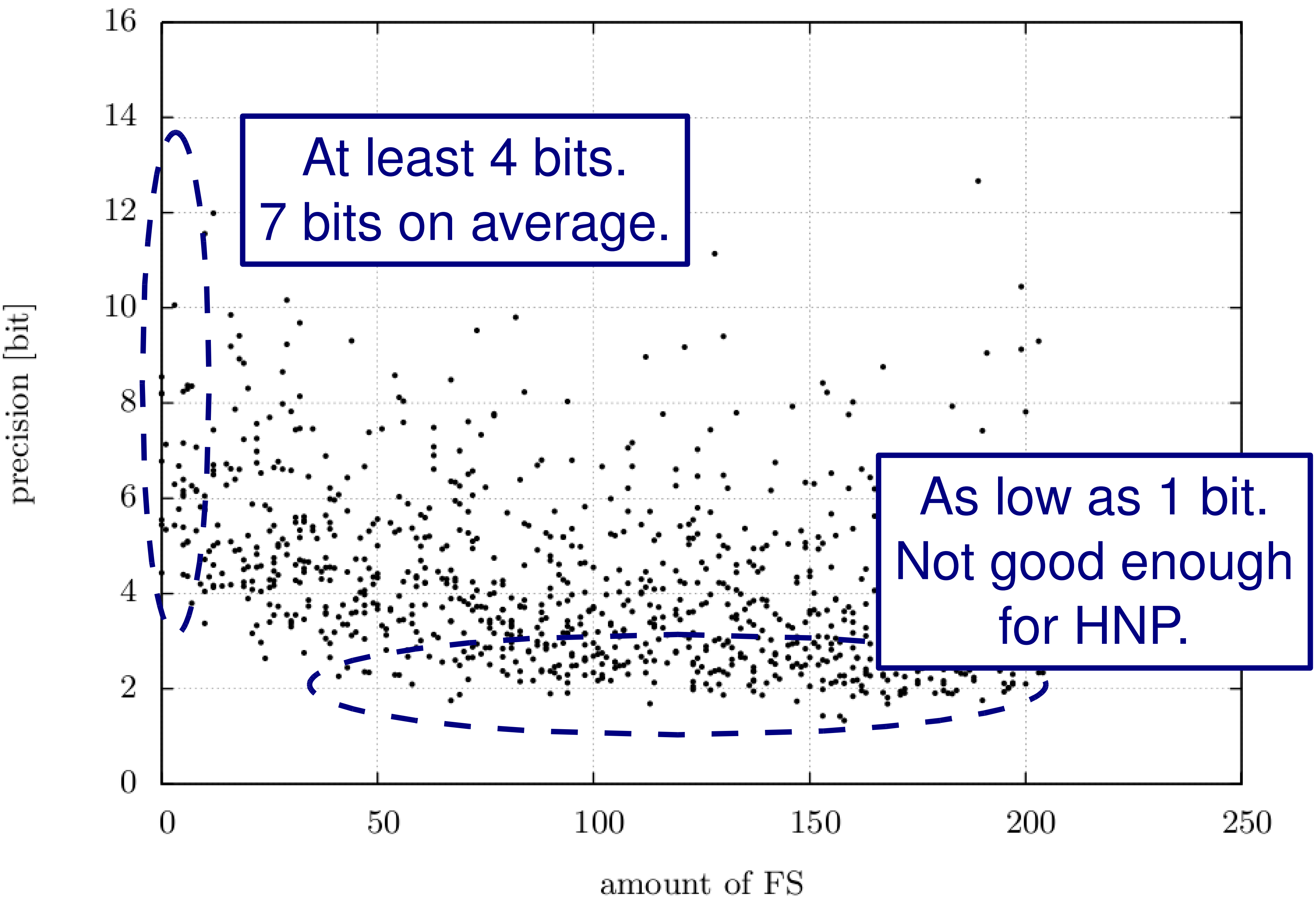
Precision of approximations II.



Precision of approximations II.



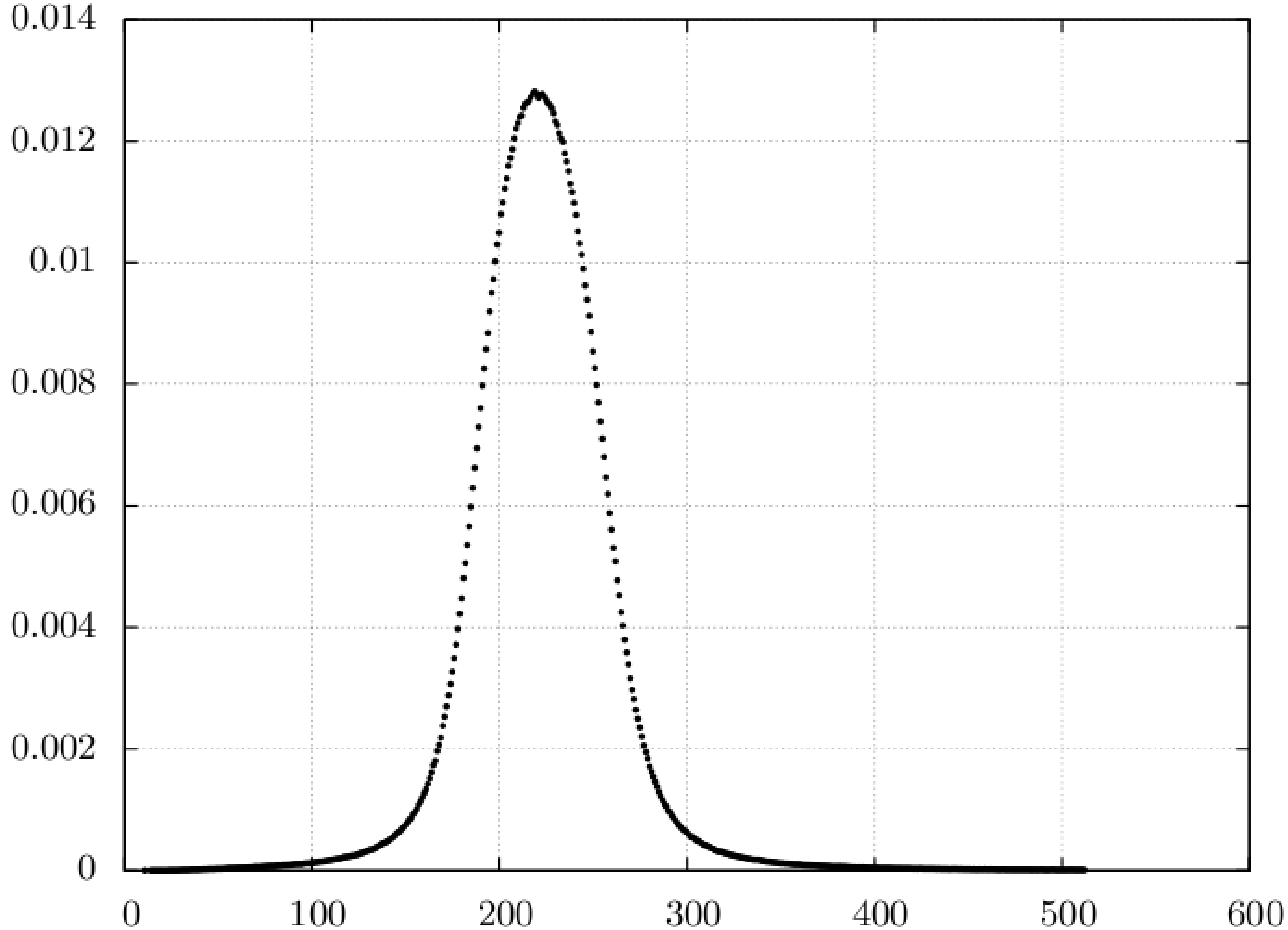
Precision of approximations II.



Tweaking n_{max}

$$\frac{m_i R \,\mathbf{mod}\, p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

- Instead of using max. number of FS, precompute ideal value



- Increases minimal precision by 1 bit

Hidden Number Problem

- Classical tool to solve modular approximations for *x*

$$|t_i x - u_i|_N < \frac{N}{2^{l+1}}$$

- Create lattice spanned by

$$\begin{pmatrix} N & 0 & \cdots & 0 & 0 \\ 0 & N & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & N & 0 \\ t_0 & \cdots & \cdots & t_{k-1} & N^{\frac{1}{2}}/2^{l+1} \end{pmatrix}$$

- Find lattice vector close to $(u_0, \ldots, u_{k-1}, 0)$
- Hope it is $\left(t_0 x - \alpha_0 N, \ldots, t_{k-1} x - \alpha_{k-1} N, \frac{x N^{\frac{1}{2}}}{2^{l+1}}\right)$

Experiments

- 5 RSA instances
- Simulated side information leakage
- 150 signatures filtered from app. 7000
  - Up to 4 final subtractions taken into account
- Minimal precision presumed from 3.5 to 8.5 bits
- <u>Factorization found in 40 minutes</u> for each instance
- Computing platform
  - 20x Opteron 844 (1.8 GHz)
  - Debian 64bit
  - NTL/GMP

Future work

- Is SCH information available in ePassport scenario?
- Other scenarios?
- Extend to pure timing attack
- Improve attack
  - Other tweaks to increase precision
  - Handle RSA blinding
- Provide proof and limits when attack works
  - probability
  - Time complexity

Conclusion

- New attack on RSA-CRT with Mont. Multiplication
- Known plaintext only (previous attacks were CCA)
- Another use of lattices and LLL algorithm
- If assumptions are true, active authentication can be broken, i.e. e-passport cloned
- Attack possible in other scenarios

Conclusion

- New attack on RSA-CRT with Mont. Multiplication
- Known plaintext only (previous attacks were CCA)
- Another use of lattices and LLL algorithm
- If assumptions are true, active authentication can be broken, i.e. e-passport cloned
- Attack possible in other scenarios

See remarks on slide 5.

Thank you for your attention! Questions?

Martin Hlaváč

Department of Algebra
Charles University in Prague

IT Security Department
Czech Insurance Corporation