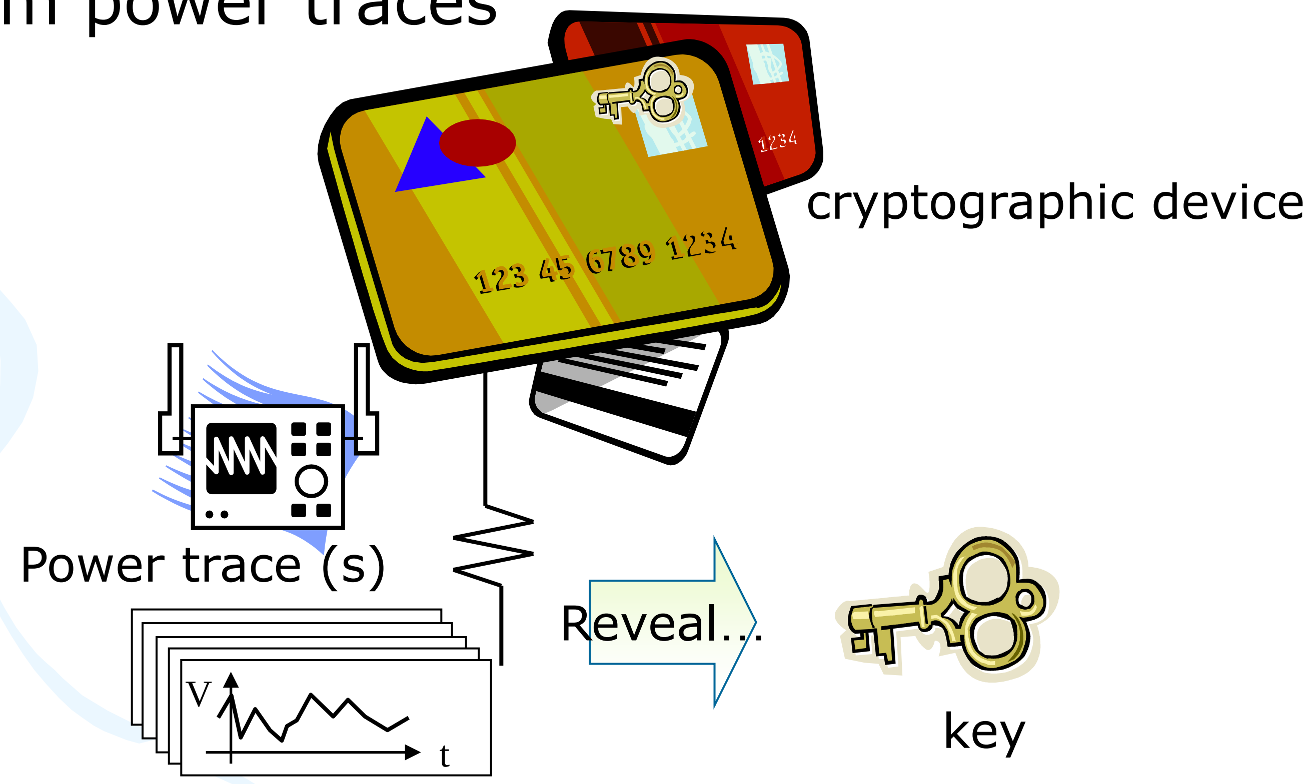
Exponent Blinding Does not Always Lift (Partial) SPA Resistance to Higher-Level Security

Werner Schindler (Bundesamt für Sicherheit in der Informationstechnik (BSI)) and
Kouichi Itoh (Fujitsu Laboratories LTD.)

7th, June, 2011, ACNS2011

Background: power-analysis

- An attack to reveal a key of a cryptographic device from power traces



- Device destruction is unnecessary
  → advantageous factor for an attacker

Bundesamt für Sicherheit in der Informationstechnik

Type of Power Analysis

FUJITSU

- Simple Power Analysis (SPA)
  - Reveals a key from a single power trace, (or from the average of single power traces to reduce noise)
- Differential Power Analysis (DPA)
  - Reveals a key by a differential of plurality of power traces

Cryptographic Devices must prevent both SPA and DPA

Attack with SPA (on RSA)

- Distinguishes elementary operations from a single power trace, which is correlated to the key bits.

RSA decryption: $c^d \pmod N$,

input: ciphertext c, private key, $d=d[k\text{-}1]||..||d[0]$ (k-bit)

$T=1$

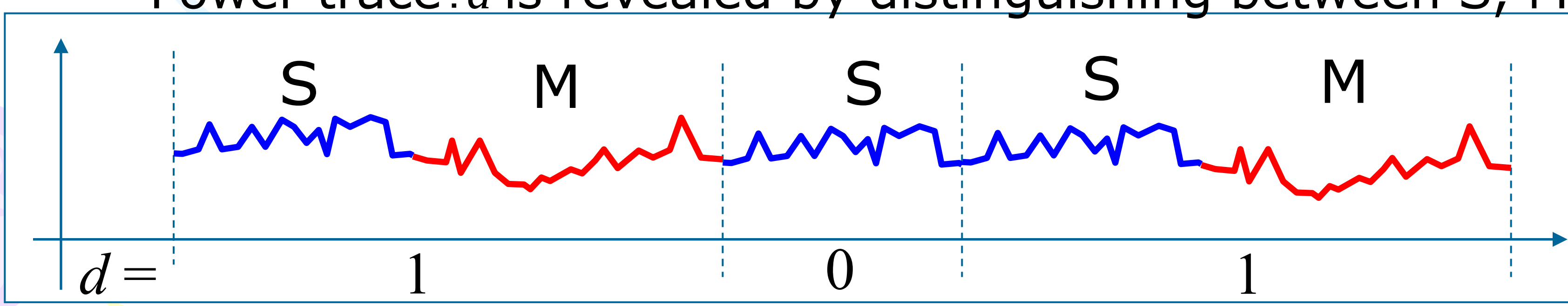
for $i=k-1$ down to 0

$T = T^2 \pmod N$ /* Square (S) */

if $\boxed{d[i]=1 \ T=T\times c \pmod N \ \text{/* Multiply (M) */}}$

return $T= c^d \pmod N$ calculated only when $d[i]=1$

Power trace: d is revealed by distinguishing between S, M



Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

SPA countermeasure

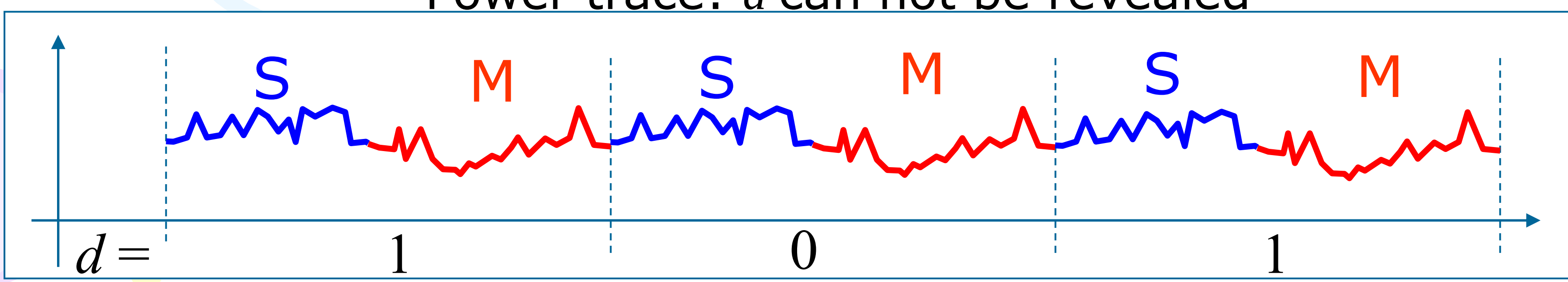
- Square and Multiply method (S&aM, Coron CHES'99)
  - Performs dummy multiplications when $d[i]=0$
  - Small-memory solution, and suitable for smartcards

RSA Decryption with S&aM: $c^d \pmod N$

input: ciphertext c, private key $d=d[k-1]||..||d[0]$
  $T[0]=1$;
  for $i=n-1$ down to 0
    $T[0] = T[0]^2 \pmod N$ /* Square (S) */
    $T[1-d[i]]=T[0]\times c \pmod N$ /* Multiply(M) */
  return $T= c^d \pmod N$

Always performed

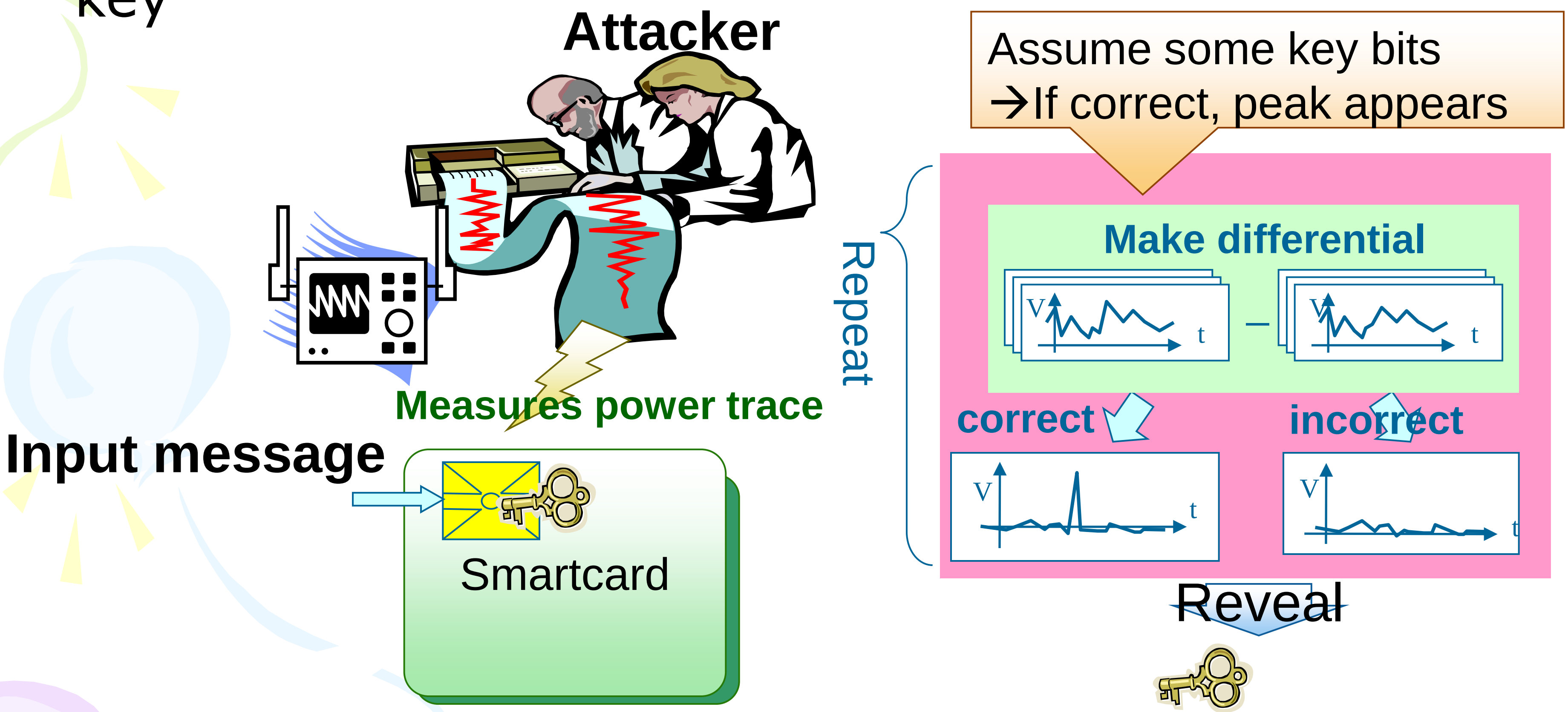
Power trace: d can not be revealed



Good solution, but could be **partial-SPA resistant** (later)

Attack with DPA

- Considers differentials of power traces to reveal the key



- <u>Data randomization</u> technique works as countermeasure

Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

DPA Countermeasure (on RSA)

- Decryption without countermeasures

$$c^d \pmod N$$

(c: plaintext, d: private key, N: modulus)

- Decryption with exponent blinding (countermeasure)

randomized

$$c^{\boxed{d+r_i\phi(N)}} \pmod N$$

(r_i: random number, $\phi(\cdot)$: Euler's totient function)

- Decryption (only) with base blinding (countermeasure)
→vulnerable to a local timing attack!(Schindler PKC'02)

randomized randomized

$$\boxed{(c(r_i^{\,e}))}^d \pmod N \times \boxed{r_i^{-1}} \pmod N$$

(r_i: random number, e: public exponent)

Bundesamt für Sicherheit in der Informationstechnik

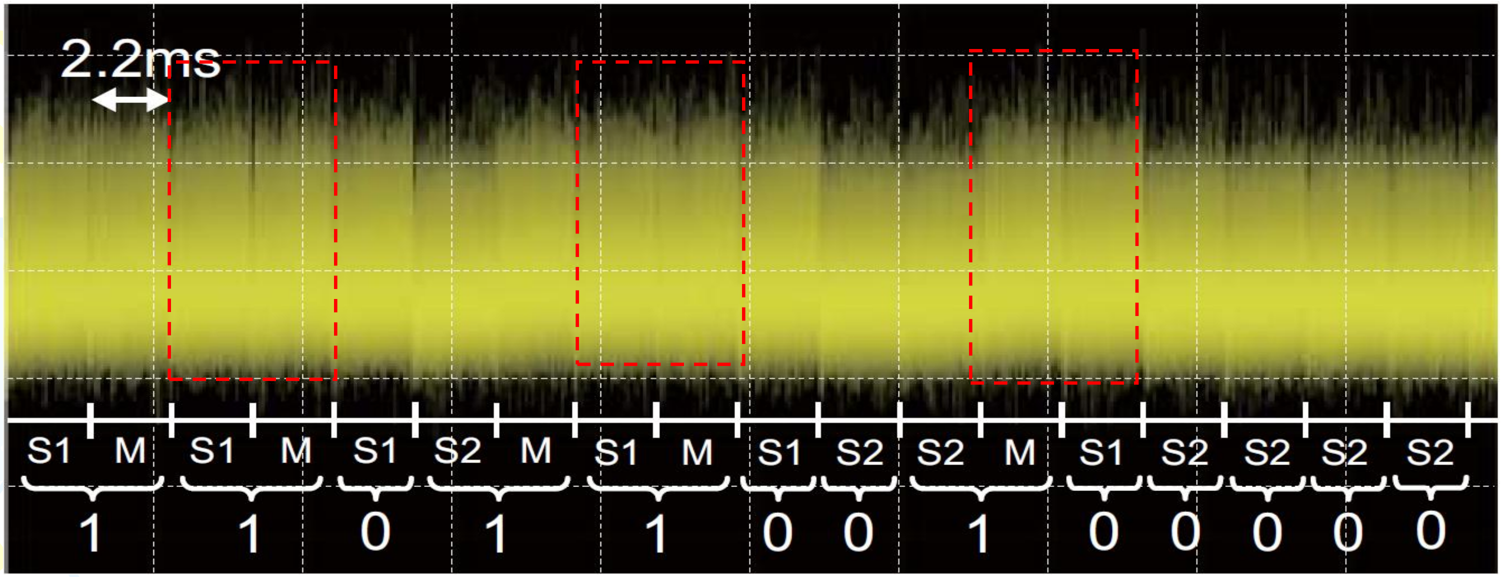
FUJITSU

Our question

- How secure is the following combination?
  - S&aM (SPA countermeasure)
  - exponent blinding (DPA countermeasure)
- We focus on its security against SPA. It is
  - secure to ‘classical’ SPA attacks, but…
  - it could be only **partial SPA-resistant** when using **special SPA attacks**

An example of Special SPA attacks (Yen, Mycrypt '05)

- By using $c = -1$, an attacker must distinguish the fixed value operation M: 1×-1, S1:$(-1)^2$ and S2:$(1)^2$



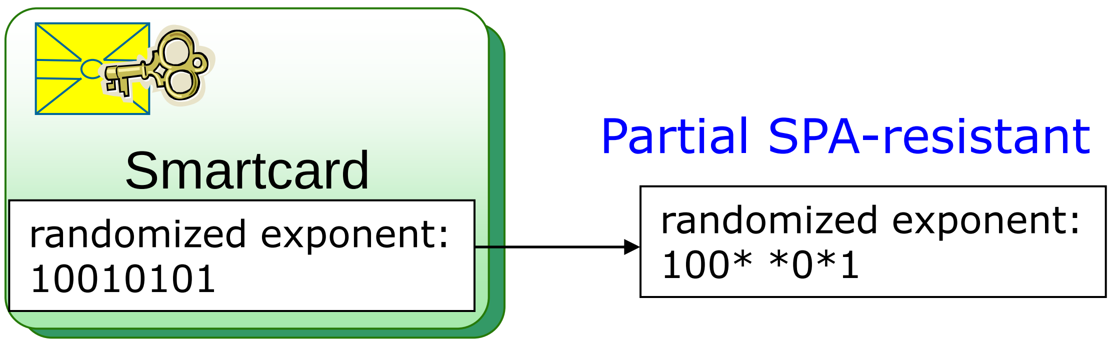
Homma et. al, "SPA using a steady value input against RSA hardware implementation", SCIS '07

- Attacks proposed by Schindler at PKC'02 and CT-RSA 2008 are also applicable

Theory: insecure and non-SPA resistant
Real: due to noise, some operations are indistinguishable
→partially SPA-resistant (some observed bits are false)

Our goal

- Partial SPA-resistance was supposed be secure enough even for **<u>very small error bits ratio</u>**
  - e.g. 5% unnoticed error bits in 1024-bit RSA key
    → recovering cost: $\sum_{j=0,..,51} {}_{1024}C_{j} = 2^{288}$



- So far:
  - Partial SPA-resistance was supposed to be secure

But...

Our result: a new attack that shows
<u>partial SPA resistance can be insufficient!</u>

Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

Our proposal

- Two attacks that tolerate error bits
  - Basic attack:
    tolerates high error rates,
    but requires many power traces and large computational workload
  - Enhanced attack:
    tolerates lower error rates than the basic attack but requires by far less power traces
- Our attacks are applicable to:
  - RSA without CRT / RSA with CRT
  - ECC

Basic attack

Notation

- v_i : randomized exponent in the cryptographic device
  - $v_i = d + r_i y$, where
    r_i: unknown random number in i-th decryption
    y: $\phi(N)$ or $\phi(p)$ in RSA, $\#E$ in ECC
- v_i' : randomized exponent with error bits, observed by an attacker
  - $v_i' = d + r_i y + e_i$, where
    r_i : random number in i-th decryption
    y : $\phi(N)$ or $\phi(p)$ in RSA, $\#E$ in ECC
    e_i: guessing error

Basic attack: Entire procedure

- Step B-1. Observe randomized exponent v_i'

  v_1' = 10010100010100101 v_2' = 00110111010101011

  v_3' = 10011100110100100 v_4' = 00111110110100011

- Step B-2. Classify v_i' with regard to the random number r_i
  - As soon as one class contains t elements it is chosen as "winning class".

Class	v_i'
1	v_1' v_3'
2	v_2' v_4' v_6'
3	v_5'

winning class

- Step B-3. Correct the error bits by applying majority decision (bitwise) to the "winning class" → key is revealed

  v_2' = 00110111010101111

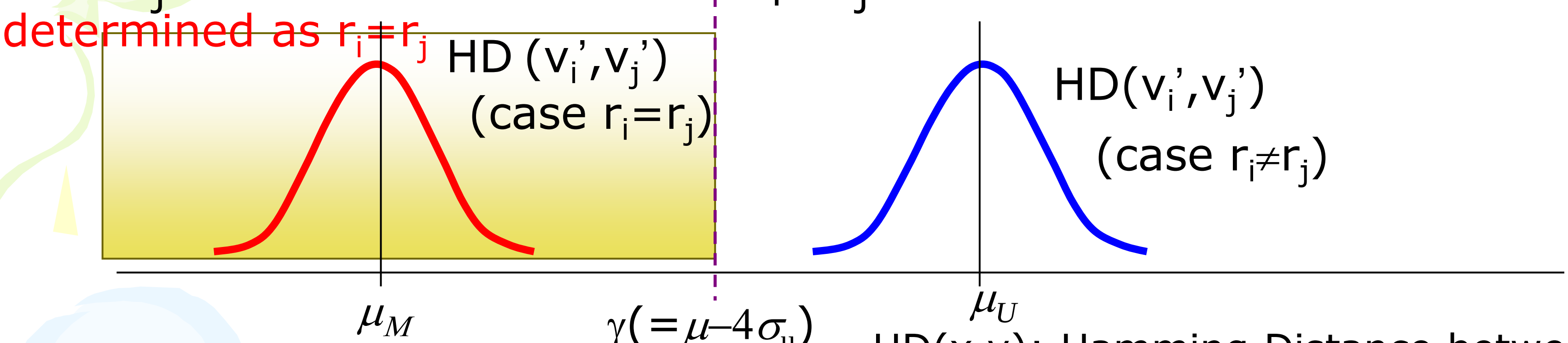
  v_4' = 00111110110100011

  v_6' = 10011110110101101

  v_i = 00111110110101111 corrected key

Step B-2:Classification of v_i'

- Use statistical test on hamming distance of v_i' and v_j' to decide whether $r_i=r_j$



HD(x,y): Hamming Distance between x,y

- Theory and experimental result shows $\leq$ 23% error in ECC, or $\leq$30% error in RSA is tolerable for classification

Experiment results on deciding whether $r_i=r_j$ (10000 trials)

ECC					RSA				
$\log_2 d$	$\log_2 r_i$	ε	TP	FP	$\log_2 d$	$\log_2 r_i$	ε	TP	FP
256	16	0.15	1.0	0.0000	1024	16	0.20	1.0	0.0000
256	16	0.20	0.9762	0.0001	1024	16	0.25	1.0	0.0001
256	16	0.23	0.8206	0.0000	1024	16	0.30	0.8756	0.0000

ε : Error ratio, TP: True Positive, FP: False Positive

Step B-3:Correct error bits with majority decision

- There is a trade-off between error rate and probability to reveal the correct key.

v_2' = 0011011101010111
v_4' = 0011111011010011
v_6' = 1001111011010110

$\left.\right\}$ t error keys

$\boxed{v_i = 0011111011010111}$ Corrected key

- Our experimental results show that for small R ≤ 23% error rate in ECC, and ≤ 28% error rate in RSA is tolerable for revealing correct key.

Experiment result on successful key revealing

ECC						RSA					
t	$\log_2 d$	R	ε	Success	#of v_i' (≥O(2^R))	t	$\log_2 d$	R	ε	Success	# of v_i' (≥O(2^R))
13	256	10	0.22	10/10	5253	17	1024	10	0.20	10/10	6890
15	256	10	0.23	09/10	7425	27	1024	10	0.25	10/10	13833
11	256	16	0.20	10/10	137000	37	1024	10	0.28	6/10	23682

ε: error ratio, R: bit length of r_i

Enhanced attack

Bundesamt für Sicherheit in der Informationstechnik

Basic attack: Disadvantages

FUJITSU

- Even moderate parameters t require
  - at least $O(2^R)$ power traces
    (R = bit length of the random numbers)
  - at least $O(2^{2R})$ computations
- NOTE: If the total number of decryptions is limited (clearly) below $2^{R/2}$:
  → Basic attack becomes definitely infeasible

To reduce the required number of power traces…

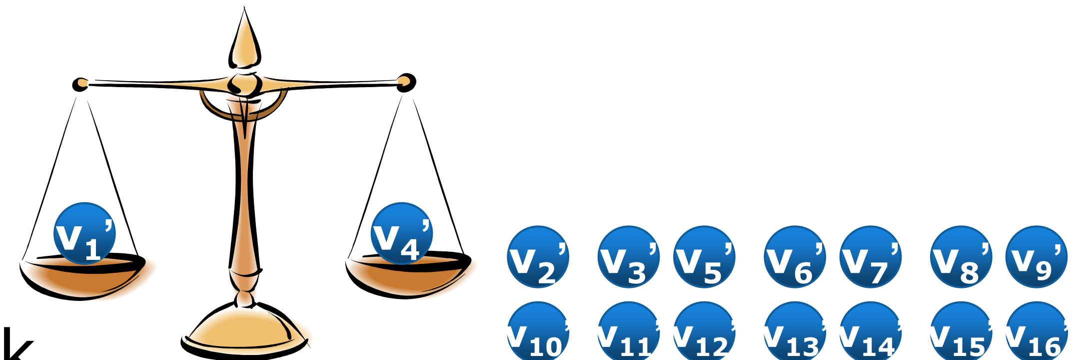
Our next proposal: enhanced attack

Bundesamt
für Sicherheit in der
Informationstechnik

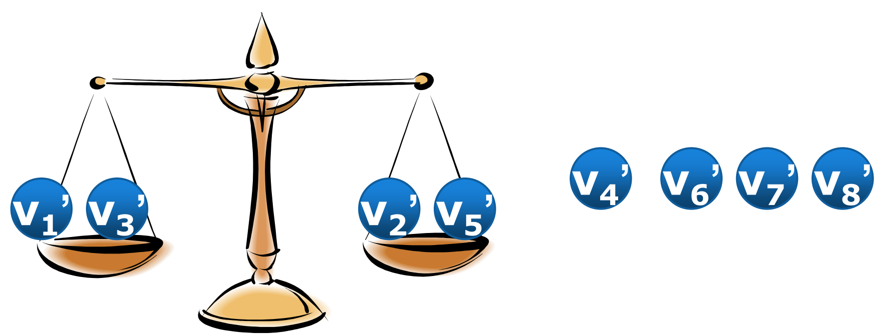
FUJITSU

New idea: u-sum algorithm

- Basic attack
  Find pairs of power traces with identical random numbers
  → Many power traces are required



- Enhanced attack
  Find pairs of u-tuples of power traces with identical sums of the random numbers (“identical u-sums”)
  → Reduces the number of power traces drastically!



Enhanced attack: Entire proc.(1/2)

- Step E-1. Observe randomized exponents v_i'

  v_1' = 1001010001010010
  v_2' = 0011011101010101
  v_3' = 1001110011010010
  v_4' = 0011111011010001

- Step E-2. Find pairs of u-tuples for which the u-sum of the random numbers r_i are equal (here: u=2)

$(v_{i,1}', v_{i,2}')$	$(v_{j,1}', v_{j,2}')$	
v_1' v_3'	v_2' v_5'	$r_1+r_3=r_2+r_5$
v_2' v_3'	v_4' v_6'	$r_2+r_3=r_4+r_6$
v_2' v_4'	v_3' v_5'	$r_2+r_4=r_3+r_5$

- Step E-3. Solve the system of linear equations obtained in Step E-2 → random numbers are revealed (up to a shift)

  r_1=100010101
  r_2=101000101
  r_3=100101010
  r_4=010101011
  r_5=111010010
  r_6=100110110

Enhanced attack: Entire proc.(2/2)

- Step E-4. Remove the random number r_i from v_i'; obtain the correct key value by majority decisions and correction algorithm

Note: In ECC, $v_i' - r_i\#E$ represents d with error bits

$v_1' - r_1\#E$ = 0011011101010101111

$v_2' - r_2\#E$ = 0011111011010000111

$v_3' - r_3\#E$ = 1001111011010101101

$v_4' - r_4\#E$ = 0010011101010101111

$v_5' - r_5\#E$ = 0111111011010001111

$v_6' - r_6\#E$ = 1011101011010101001

(simplified description)

d = 0011111011010101111

Corrected key

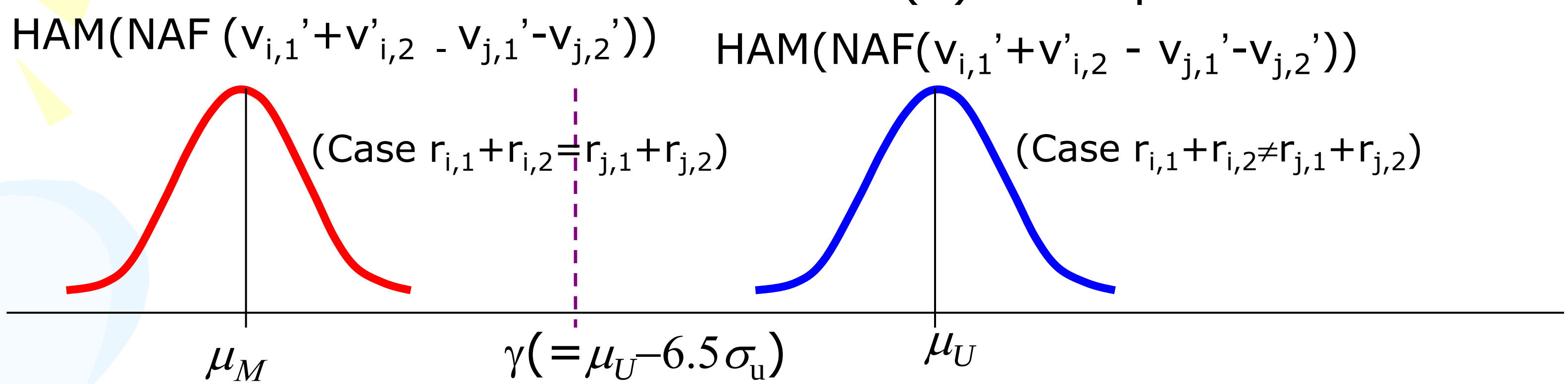
(*) For RSA, attacker tries to obtain $\phi(N)$ in place of d.

Unlike for the basic attack
all v_i's can be used for majority decision

Step E-2:Finding u-tuple pairs

- Apply statistical test on Ham(NAF($v_{i,1}'+v_{i,2}'$ - $v_{j,1}'-v_{j,2}'$)) to decide whether $r_{i,1}+r_{i,2}=r_{j,1}+r_{j,2}$

(*)NAF representation is used.



- Theoretical and experimental results show:
  - In 256-bit ECC, 8% (u=2), 6% (u=3), or 4% (u=4), and
  - in 1024-bit RSA, 13% (u=2), 9% (u=3), or 6% (u=4)

  are tolerable for 16-bit random numbers r_i.
- The attack efficiency decreases for increasing R but scales much better than the basic attack

Step E-3:Solve linear equations

Obtained equations from Step E-2

$r_1+r_3=r_2+r_5$
$r_2+r_3=r_4+r_6$
$r_2+r_4=r_3+r_5$

$$\Rightarrow \quad B\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- With enough equations dim(ker(B))=2
  → We proved that a basis of ker(B) is given by
  - (1,1,..,1) and
  - $(r_1,\ldots,r_N)$ = correct random numbers used by the device

⇓

r_i's are revealed (up to an additive shift)

Step E-3:How many linear equations/power traces are required?

- Theory

$$E(\#linear\ equations) \approx \frac{N^{2u}}{2u!u!} \times \frac{c(u)}{2^R}$$

where N: number of power traces
R: bit length of the random number

$$c(2) \approx \frac{2}{3}, c(3) \approx \frac{11}{20}, c(u) \approx \frac{\sqrt{3}}{\sqrt{\pi u}} (u \geq 4)$$

- Experiments

	u=2		u=3		u=4	
N	116	128	28	32	16	20
rate on dim(ker(B))=2	43/50	49/50	49/50	50/50	12/50	50/50

Estimated number of power traces (theory)

#power traces is reduced to 128 (u=2), 32 (u=3) or 20 (u=4)

Step E-4:Experimental results

- ECC ($\log_2 d=256$, $\log_2 r_i=16$, 300 trials)

ε	N	success rate
0.13	**25**	**99.7%**
0.13	**30**	**100%**
0.08	16	100%
0.08	10	91%

- RSA ($\log_2 d=1024$, $\log_2 r_i=16$, 300 trials)

ε	N	success rate
0.13	**100**	**99%**
0.13	**128**	**100%**
0.08	45	95%
0.08	35	74%

ε: error ratio

Tolerates 13% error in both ECC and RSA

Summary and conclusion

Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

Summary

- Comparison of the proposed attack

	u	R	error rate		# of power trace	
			ECC	RSA	ECC	RSA
Basic	_	10	23%	28%	7425	23682
	_	16	20%	maybe ~26%	137000	maybe as ECC
Enhanced	2	16	8%	13%	128	128
	3	16	6%	9%	32	32
	4	16	4%	6%	20	20

- Countermeasure
  - Using large random numbers
  - $\geq$ 64-bit random numbers should suffice

Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

Conclusion (I)

- We proposed two novel attacks that can break S&aM (SPA countermeasure) combined with exponent blinding (DPA countermeasure), even when the observed exponents include **error bits**
  - Basic attack:
    principally tolerates higher error rates ($\geq 20\%$), but even moderate t's require ($\geq O(2^R)$) power traces and many comparisons ($\geq O(2^{2R})$);
    Attacks on $R \leq 24$ (probably also for larger R) should definitely be feasible.
    If the number of power traces is (significantly) smaller than $2^{R/2}$ even 2-birthdays will not occur.

Bundesamt für Sicherheit in der Informationstechnik

FUJITSU

Conclusion (II)

- Enhanced attack:
  tolerates lower error rates ($\leq$ 13%),
  but requires only a small number of power traces
  ($\leq$ 128 for R=16)
  Attacks on R $\leq$ 40 (probably also for larger R) should definitely be feasible.

- We showed the effectiveness of our attack both theoretically and experimentally
- For increasing R the efficiency of both attacks decreases but the enhanced attack scales much better

Thank you for your attention!