Federal Office for Information Security

Royal Holloway University of London

Optimal Recovery of Secret Keys from Weak Side Channel Traces

Werner Schindler — BSI, Germany — Werner.Schindler@bsi.bund.de

Colin D. Walter — RHUL, UK — Colin.Walter@comodo.com

Outline

- The problem, background & history
- A typical randomised exponentiation algorithm
- The optimal decision strategy
  - General strategy and the Main Theorem
  - Example
- A metric to measure fitness of recoding guesses
- Results
- Conclusion

Background

Several standard SW measures to counteract
Side Channel Leakage from Exponentiation:

1. Blind the exponent by adding a random multiple of the group order.
2. Pick an algorithm where the pattern is independent of the secret exponent, e.g.
   - Square-and-always-multiply
   - Montgomery Powering Ladder
3. Use an algorithm where the pattern is randomised:
   - Liardet-Smart
   - Oswald-Aigner
   - Ha-Moon
   - Mist

We look at the strength of (3) with Ha-Moon for examples.

Problems for an Attacker

- There is always a lot of noise in measurements.
- Averaging to determine correct key bits is essential.
- For randomised exponentiation algorithms,
  Square & Multiply operations
  cannot be aligned directly with key bits.
- Incorrect bit deductions will always occur.
- The locations of likely errors must be identified
  for a computationally feasible algorithm.

Example: Ha-Moon

Recode the binary representation of key K from right to left:

- Add in the Carry of 0 or +1 to give new K.
- Choose digit 0 if K even.
- Randomly choose digit ±1 if K odd.
- Set Carry to be 1 for digit –1, otherwise 0, & shift K down.

Exponentiation M^K in ECC:

- Repeatedly:
  - i) read next digit (from left to right)
  - ii) perform point double
  - iii) do point add if $d = 1$ or point subtract if $d = -1$.

Traces may have different lengths: the ith operation is associated with different bits in different traces.

Ha-Moon (II)

Here are some recodings of $32n+13$:

(**D** = double, **A** = add)

… 0 1 1 0 1	… D D A D A D D A
… 0 1 1 1 –1	… D D A D A D A D A
… 1 0 0 –1 –1	… D A D D D A D A
… 1 0 –1 0 1	… D A D D A D D A
… 1 0 –1 1 –1	… D A D D A D A D A
… 1 –1 1 0 1	… D A D A D A D D A

Aim: to recover *K* from leakage like the above.

The average operator yields almost no information:
data from the top bit gets spread over several columns.

History

Karlov & Wagner (CHES 2003)
Green, Noad & Smart (CHES 2005)

- Uses a Hidden Markov Model
- Applies Viterbi's algorithm to find the best fit key.
- Treats traces serially one by one
- Convergence is unlikely with weak leakage – it can't get started.

Walter (CHES 2008)

- Restructured to process traces in parallel, and bits serially
- Better convergence on weak leakage
- Lack of sound theoretical justification

Schindler (PKC 2005)

- Optimal decision strategy identifying most likely key
- Computationally infeasible in this context

A Formal Approach (I)

Set of admissible keys:

- $\mathcal{K} \subseteq \mathrm{F}_2{}^*$

Set of all possible recoding sequences:

- $\mathcal{R} \subseteq \mathcal{D}^*$ where $\mathcal{D}$ = {admissible recoding digits}

Strategy (generic description):

- For each power trace pow_j ($1 \leq j \leq N$) guess the individual recoding digits, yielding (disturbed, possibly invalid) noisy recoding sequences $G_1, \ldots, G_N \subseteq \mathcal{D}^*$.
- Select the key K^* that fits $G_1, \ldots, G_N$ best.

A Formal Approach (II)

Interpretation of the noisy recoding sequences as a two-step random experiment:

- j^{th} randomised recoding sequence:

$$\varphi: \mathcal{K} \times \mathcal{Y} \rightarrow \mathcal{R}, \quad \varphi(K, y_j) := R_j$$

where y_j is a random number. The target device contains a Finite Automaton to do this.

- j^{th} noisy recoding guess:

$$\psi: \mathcal{R} \times \mathcal{Z} \rightarrow \mathcal{D}^*, \quad \psi(R, z_j) := G_j$$

where z_j is a random number. The result of the adversary's inaccurate measurements.

Main Theorem (I)

Theorem 1(ii) (a special case):

Assumptions:

- The unknown key K has been selected randomly according to some probability distribution η
- Given recoding sequence guesses $G_1,\ldots,G_N$.
- The adversary can detect whenever an operation of the recoded sequence R is carried out and guesses the types of these operations independently.

Notation: $p(g|r) :=$
Prob(guessed op^n type is g given the true op^n type is r)

Main Theorem (II)

Theorem 1(ii) (special case, ctd'):

The optimal decision strategy selects a key $K^* \in \mathcal{K}$ that maximises the term

$$\sum_{j=1}^{N} \log\Big(\sum_{\substack{R \in R(K): \\ len(R)=len(G_j)}} \prod_{i=0}^{len(G_j)-1} p(g_{j,i} \mid r_i) \Big)$$

assuming keys and recodings are distributed uniformly.

<ins>Note:</ins> Theorem 1(i) in the paper treats the most general case.

Traces

The side channel gives a sequence of probabilities that the underlying operations correspond to particular digits.

- So we define a trace by $T = (t_i)_{\{0 \le i < \mathrm{len}(G)\}}$
  with t_i = probability distribution on $\mathcal{D}$
  (depending on the power trace)
- Thm 2 (a corollary of Thm 1 for traces) enables us to replace $p(g_j|r)$ by t_i and so avoid guessing recodings G.

Example (I)

Application of Theorem 1(ii):

Ha-Moon recoding with artificially small parameters:

- $\mathcal{K} = \{0,1\}^n \setminus \{(0,\ldots,0)\}$, $\mathcal{D} = \{\text{'S'}, \text{'M'}, \text{'}\overline{\text{M}}\text{'}\}$

Stochastic simulations

- – Select K randomly
- – Generate N recoding sequences $R_1,\ldots,R_N$
- – Generate N noisy recoding sequences $G_1,\ldots,G_N$ by flipping recoding digits randomly

- More precisely:

$p(\text{'M'} \mid \text{'S'}) = 0.2$, $p(\text{'}\overline{\text{M}}\text{'} \mid \text{'S'}) = 0.1$

$p(\text{'}\overline{\text{M}}\text{'} \mid \text{'M'}) = 0.2$, $p(\text{'S'} \mid \text{'M'}) = 0.1$

$p(\text{'M'} \mid \text{'}\overline{\text{M}}\text{'}) = 0.2$, $p(\text{'S'} \mid \text{'}\overline{\text{M}}\text{'}) = 0.1$

Example (II)

Application of Theorem 1(ii):
Ha-Moon recoding with artificially small parameters:

- 100 stochastic simulations per table row

The correct key was ranked

Key length	# traces	1st	2nd	3rd - 9th	10th - 99th	100- 999	>1000
15	10	84	5	9	1	1	0
20	10	57	20	20	2	1	0

Example (III)

Application of Theorem 1(ii): Ha-Moon recoding with artificially small parameters:

- These numerical results are remarkable since each recoding digit is correctly recognised despite probability of only 70% for individual operations!

However,

- unlike in many other side-channel attacks the optimal decision strategy cannot be applied to small portions of the key.
- Hence the application of Theorem 1 is infeasible for real-world key parameters.
- Starting from the optimal decision strategy a computationally feasible approximator is derived.

Federal Office for Information Security

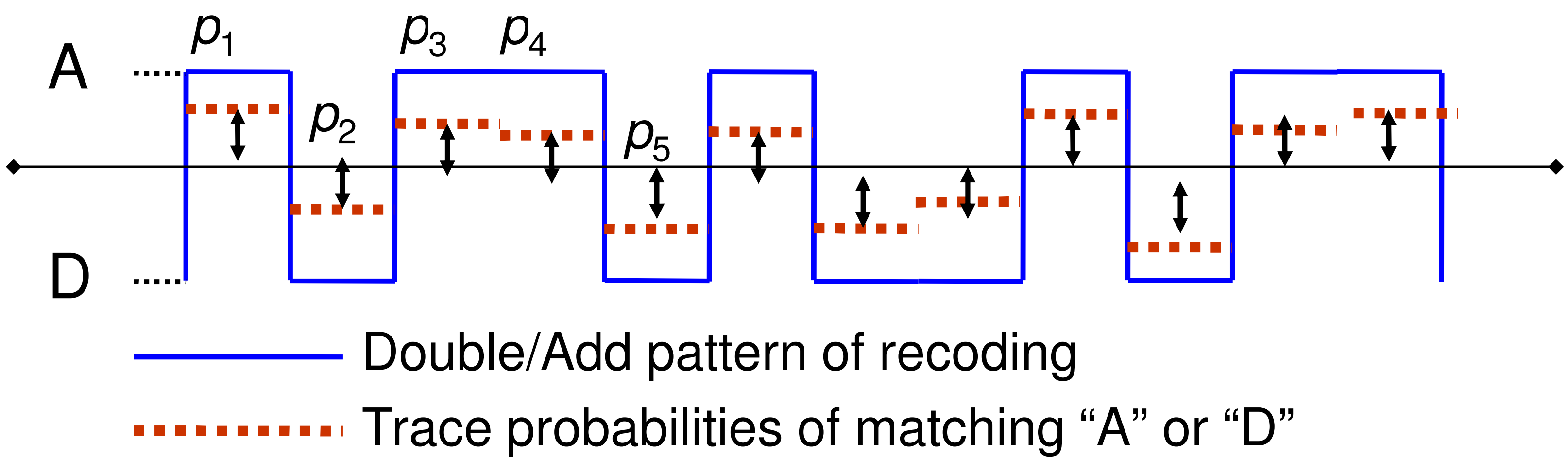
Royal Holloway University of London

The Metric (I)

Replace CHES08 distance between a trace *t* and recoding *r*

$\mu(t,r) = \sum_i (1-p_i)$ by "credibility" $\mu(t,r) = \prod_i p_i$

where p_i is probability that the i^{th} operation in *t* is the same as the i^{th} operation for *r*. (Hamming dist. vs Prob^y.)



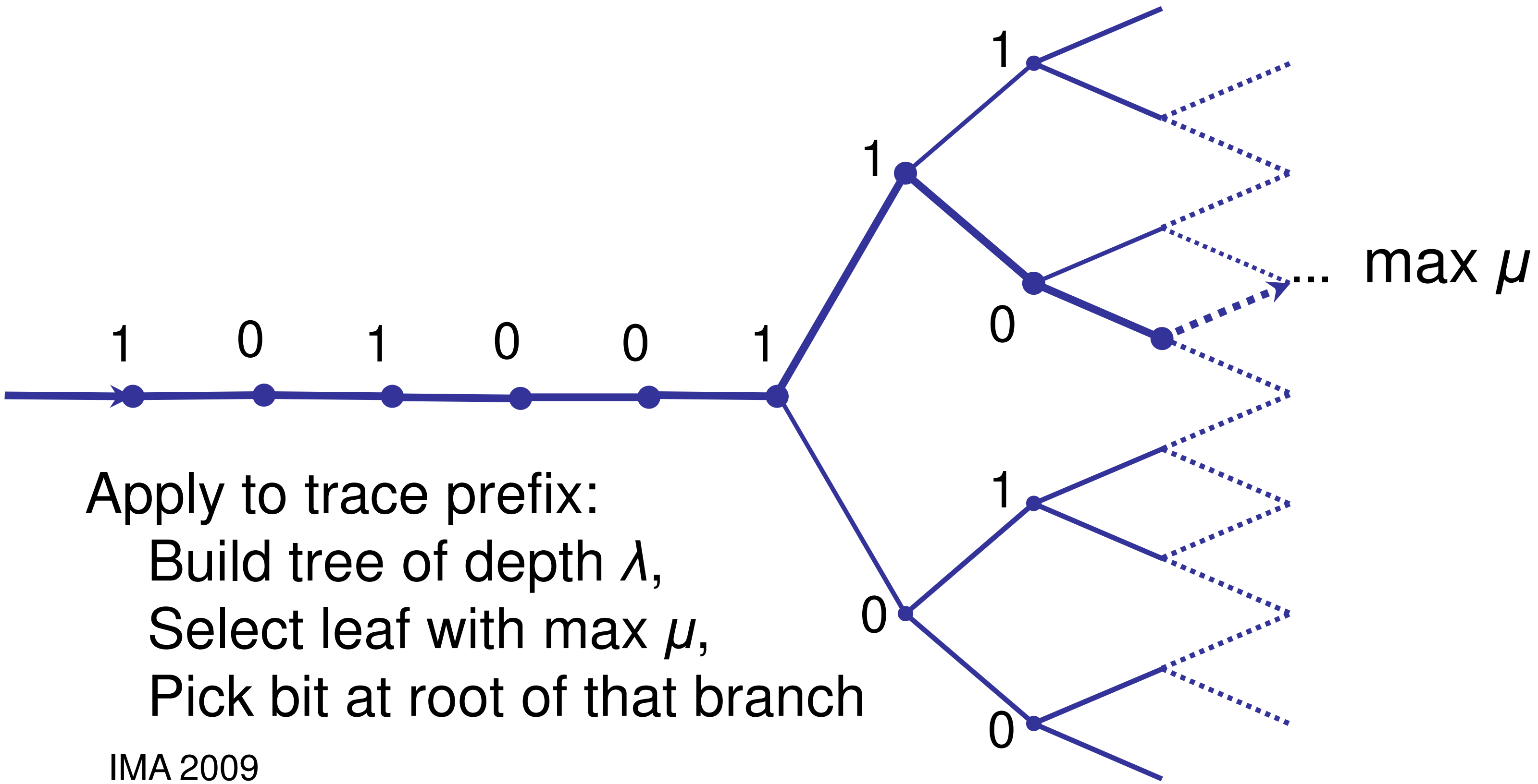
$\mu = p_1 \times p_2 \times p_3 \times p_4 \times \ldots$

The Metric (II)

- Define the ***credibility*** of key choice K for a trace t by

$$\mu(t,K) = \sum_r \{\ \mu(t,r) \mid r \text{ is a recoding of } K\ \}$$

This selects the best match recoding of K.



Apply to trace prefix:

Build tree of depth λ,

Select leaf with max μ,

Pick bit at root of that branch

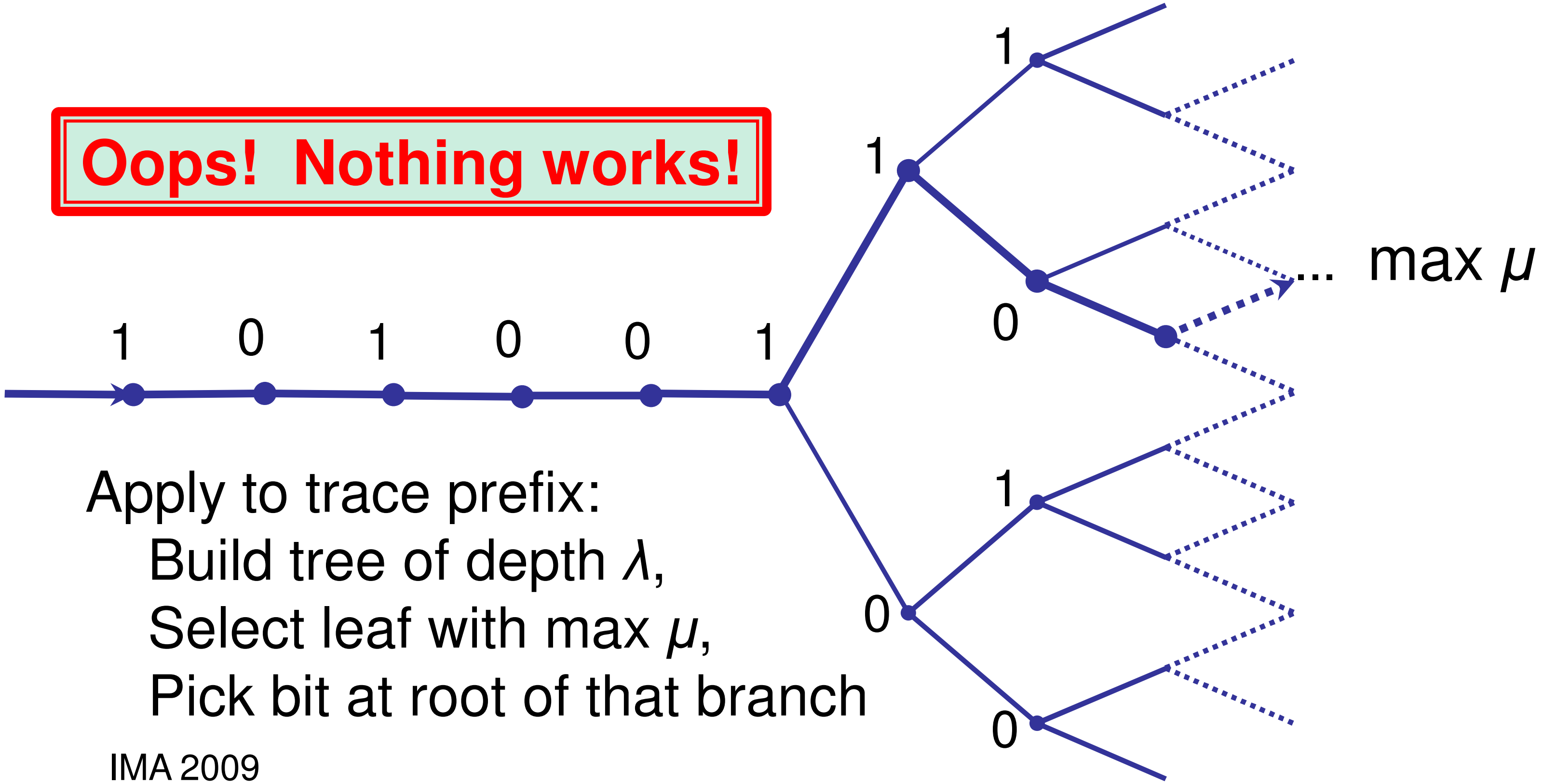
The Metric (II)

- Define the ***credibility*** of key choice K for a trace t by

$$\mu(t,K) = \sum_r \{ \mu(t,r) \mid r \text{ is a recoding of } K \}$$

This selects the best match recoding of K.

Oops! Nothing works!



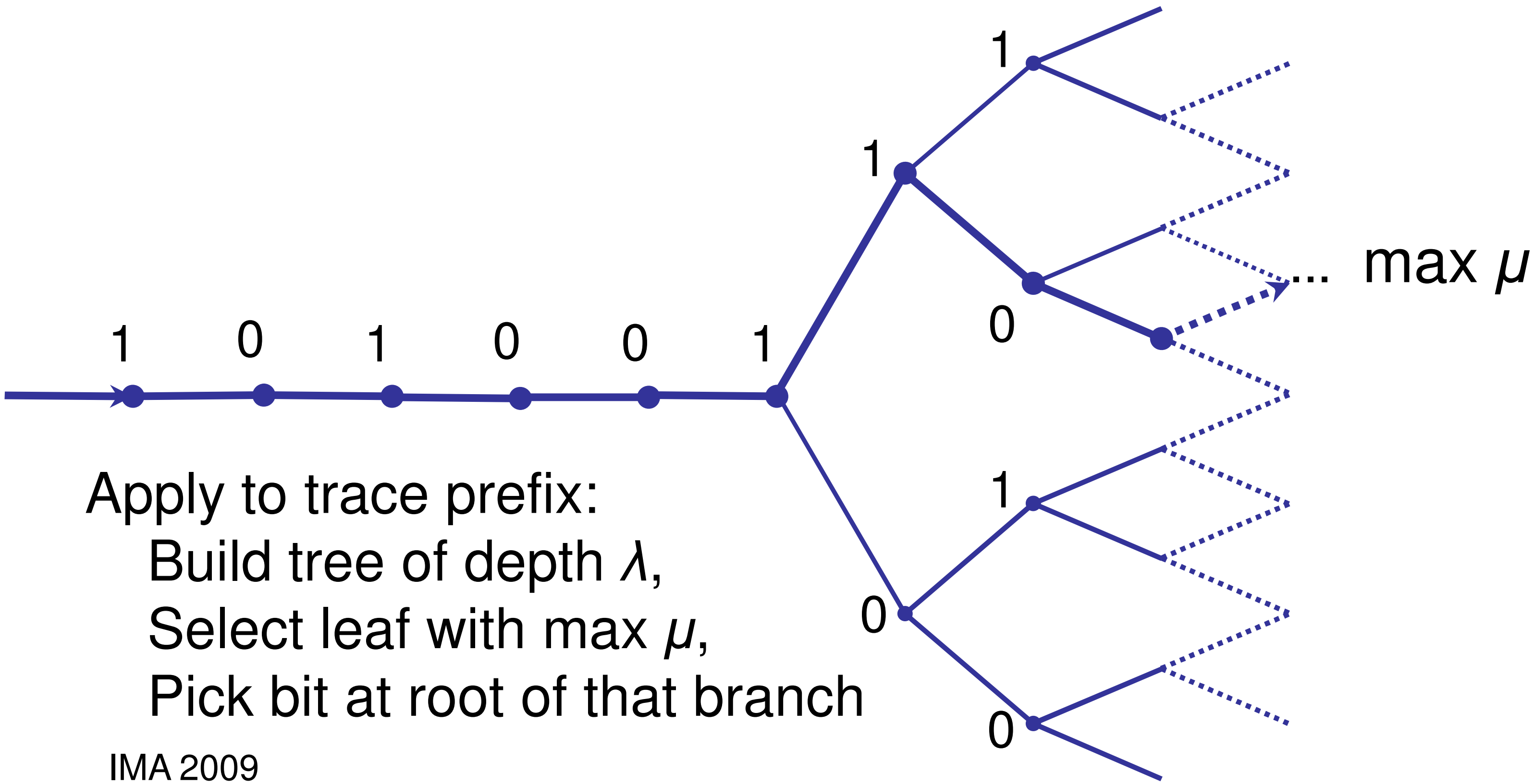
Apply to trace prefix:
Build tree of depth λ,
Select leaf with max μ,
Pick bit at root of that branch

The Metric (III)

- Modify the credibility definition, replacing "sum" by "max":

  μ(*t*,*K*) = max { μ(*t*,*r*) | *r* is a recoding of *K* }

  to select the best match recoding of *K.*



Apply to trace prefix:
Build tree of depth *λ*,
Select leaf with max *μ*,
Pick bit at root of that branch

Federal Office for Information Security

Royal Holloway University of London

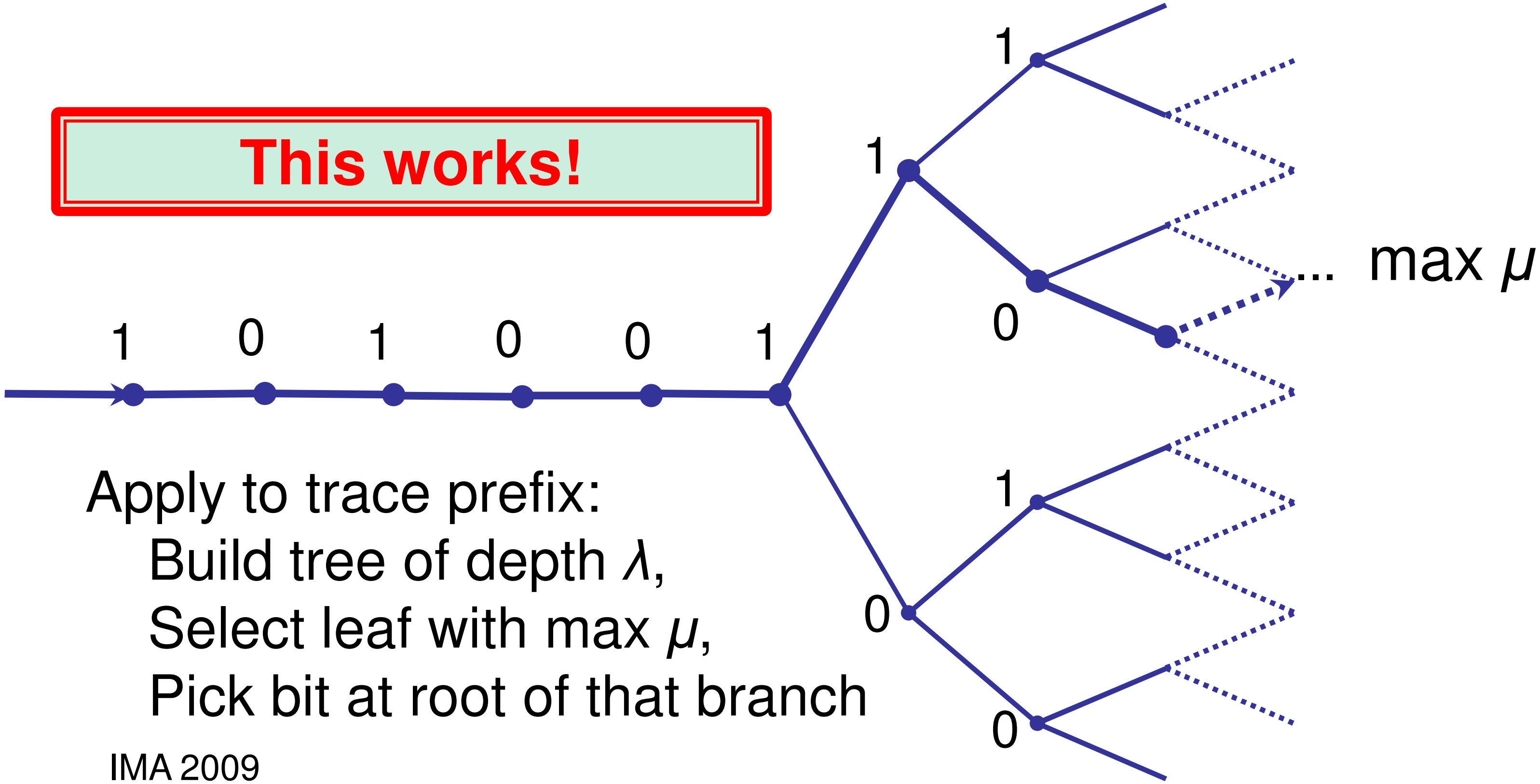
The Metric (III)

- Define the ***credibility*** of key choice *K* for a trace *t* by

$$\mu(t,K) = \max \{ \mu(t,r) \mid r \text{ is a recoding of } K \}$$

This selects the best match recoding of *K*.

This works!



Apply to trace prefix:
Build tree of depth *λ*,
Select leaf with max *μ*,
Pick bit at root of that branch

The Metric (IV)

1. Define the *credibility* of a recoding *r* for trace *t* by

   $$\mu(t,r) = \prod_i p_i$$

   where p_i is probability that the *i*th operation in *t* is the same as the *i*th operation for *r*.

   This should be large for correct interpretation of the trace.

2. Define the *credibility* of a key choice *K* for trace *t* by

   $$\mu(t,K) = \max \{ \mu(t,r) \mid r \text{ is a recoding of } K \}$$

   to select the best match recoding of *K.*

3. Define the *credibility* of a key *K* for trace set *T* by

   $$\mu(T,K) = \sum_{t\in T} \log(\mu(t,K)) \text{ or } \sum_{t\in T} \mu(t,K)$$

   The best fit key maximises this. (The latter is slightly better.)

Properties

- Traces become aligned correctly (or almost correctly) with key bits/digits by selecting the best fit recoding.
- Summing the metric values for best recodings of each trace provides the averaging that reduces noise and enables the best key bit to be selected.
- Locations for incorrect bits can be determined by looking at the difference in the credibility of the 0- and 1- branches of a node in the tree. A *small* difference means lack of certainty about the decision.
- Key bit positions can be ordered according to this probability of correctness.

Some Figures

- Take the Ha-Moon 1 recoding.
- Assume a 70% chance of deciding correctly between a square or multiplication from the side channel trace, but *unable* to distinguish the multiplications for –1 and +1.
- Take typical 192-bit ECC key & *only 5 traces*.
- On average there are only 20.7 bit errors
- In 1.3% of cases there are *no* errors in the 168 bits we are most certain of, leaving just 24 *known* bits to check.
- *It is computationally feasible to correct all errors in these.*

Federal Office
for Information Security

Royal Holloway
University of London

Conclusion

- Traces from randomised exponentiation algorithms can be aligned effectively to pool weak side channel leakage associated with individual key bits.
- Locations of possible bit errors are identified with ease, making it computationally feasible to correct them.
- Theoretical results on the optimal decision strategy were applied to redesign a previous algorithm for this.
- A more successful algorithm resulted, with sounder basis and better understanding of good parameters to choose.