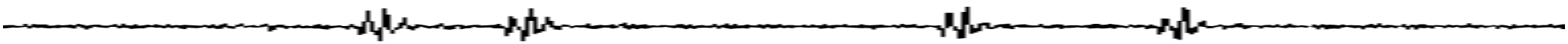
Do we need a theory for side channel attacks?

Elisabeth Oswald
University of Bristol

Motivation

- What is theory ??
  - Mathematical formalism vs. reasoning vs. providing evidence vs. falsifying
- What can it do for us ??
  - Write statements clearly
  - Make claims that can either be proven mathematically or falsified by some experiment
- Where does it most impact ??
  - Probably in the areas of evaluation/testing

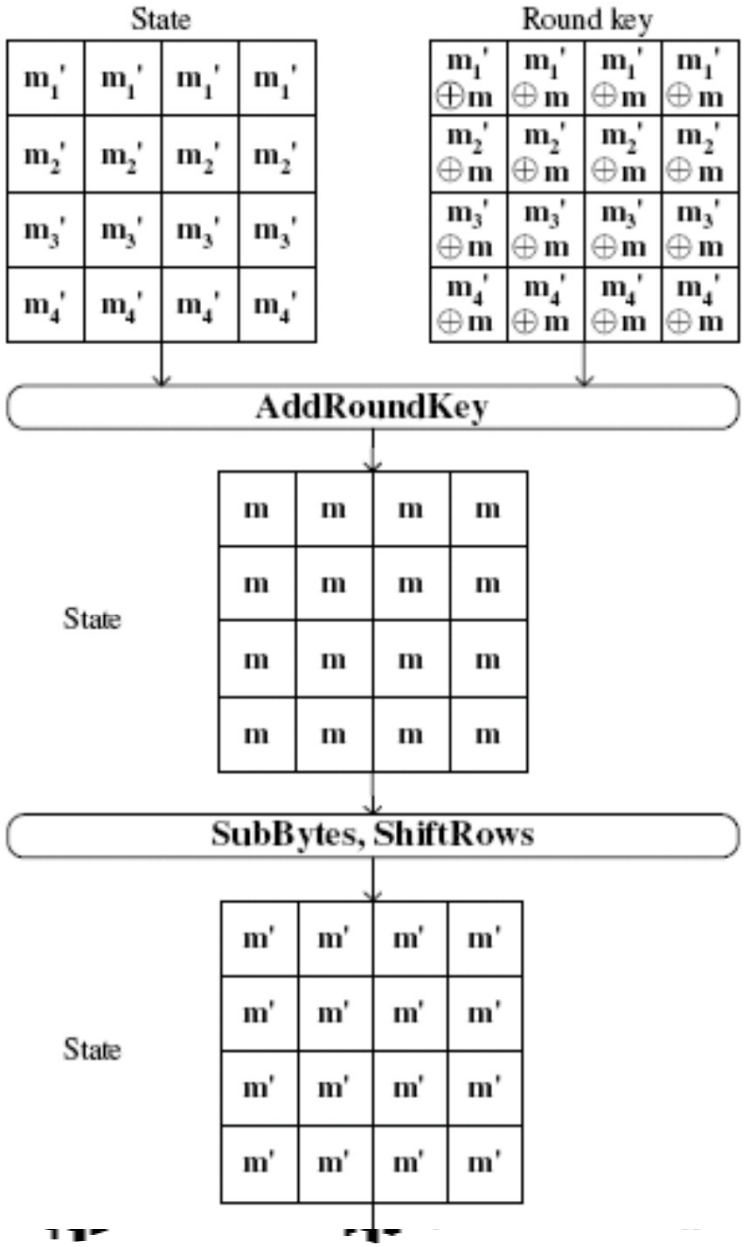


Outline

- Example 1: masking schemes
  - What kind of theory
  - How it is used
  - How does it relate to practice
- Example 2: distinguishers
  - Like for like comparison
  - Precise definition: ‘generic’
- Example 3: leakage detection
  - Like for like comparison

Provably secure masking schemes

- Masking refresher:
  - Want to conceal any/all intermediate values by random values (these are called masks)
    - Picture shows how such masks may be introduced in SW
- AES example:
  - ARK: linear operation (⊕), used to re-mask values
  - SB: $S_m(v \oplus m) = S(v) \oplus m'$, mask changed m → m'
  - SR: all state bytes use the same mask, no change
  - MC: requires 2 or more masks to avoid unmasking



Provably secure masking (historically)

- Need to define 'conceal' mathematically
  - Unpredictable (given leakage)? Too strong
  - Indistinguishable (given leakage)? Too strong
- Need to put potential definition in context with concrete attack
  - DPA style adversary requires to predict intermediate values for several traces
    - Secure against DPA if attacked intermediate values are distributed uniformly at random OR **have same distribution irrespective of (plaintext, key)**
    - **This really is a necessary condition for security but it is not sufficient**

Provably secure masking, cont.

- So prove that distribution of any intermediate value is independent of the inputs
  - How do you specify intermediate values
    - Practice shows that, e.g. in hardware implementations this is not fully achievable
- But is that a security proof?
  - Not in the sense that it gives no guarantee about the 'entire' cipher/implementation, i.e. the composition of operations that happens in practice is outside of the scope of the proof
    - We all know about the problem with glitches in hardware and that it renders masking almost entirely useless

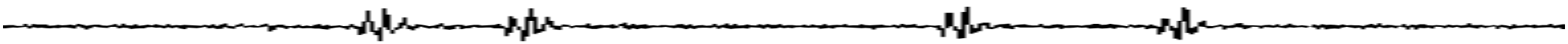
Breaking provably secure masking

- But what about a 'good' software implementation?
  - Assume first or second order masking, and that great care was taken so that there is no first order leak in encryption rounds
  - Remember our previous proof is concerned with isolated intermediate values only, so e.g. it says nothing about any leakage of a sequence of using the same mask in an encryption round
    - Clearly this means HO attacks are not dealt with
- Prime target: the computation of the Table S_m
  - $S_m(v \oplus m) = S(v) \oplus m'$
    - I call m the adress mask and m‘ the data mask

Breaking PS masking, cont.

Focus our attention hence on the computation of the masked table

- Either on the fly (unlikely to happen in practice as it is very inefficient), or prior to the encryption
- Either way this leads to a 'nice' easily identifiable pattern in a power trace
- Assume that it hasn't just been implemented naively but with some randomisation
  - Random start index
  - LFSR based random walk
  - (small) permutation



Breaking PS masking, cont.

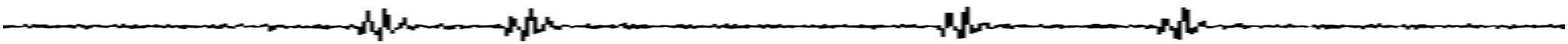
Only an unrealistically large permutation choice prevents this attack, this conclusion holds also for arithmetic masking and second order Boolean masking. See a forthcoming paper by Tunstall et al., FSE 2013.

Data Mask Recovery Rates, Boolean Masking, ARM 7

Error (bits)	0	1	2	3	4	5	6	7
RSI	0.94	0.035	0.004	0.006	0.008	0.004	0	0
RW	0.35	0.52	0.11	0.011	0.004	0.002	0.001	0
P4	0.84	0.093	0.017	0.016	0.013	0.012	0.007	0
P8	0.47	0.15	0.11	0.066	0.10	0.061	0.03	0
P16	0.0064	0.11	0.19	0.23	0.21	0.12	0.065	0.01
P32	0.011	0.052	0.13	0.25	0.27	0.19	0.081	0.01

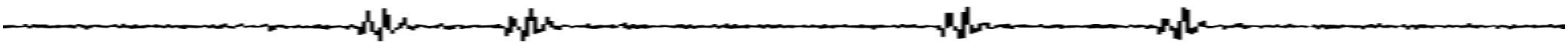
Breaking PS masking, cont.

- Clearly the gap in the 'proof' implied such an attack is conceivable
  - The proof only covered an isolated intermediate value
  - An actual proof for a secure masking scheme w.r.t. an attack of order d would need to argue that any combination of up to d values would not leak enough information
  - Masking against Side Channel Attacks: a Formal Security Proof, Prouff & Rivain, forthcoming (Eurocrypt 2013)



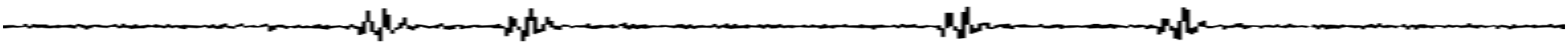
Distinguishers ...

- Now which statistic am I going to choose for my attacks today ….?
- Lot's of options, lot's of opinions, but are there any 'hard facts'?
- We need some method to make like-for-like comparisons
  - Experiments are a good way initially but we need to be wary of limitations when working with 'real' data
    - Estimation, what is actual power model, noise margin
  - We want to compare distinguishers not devices!



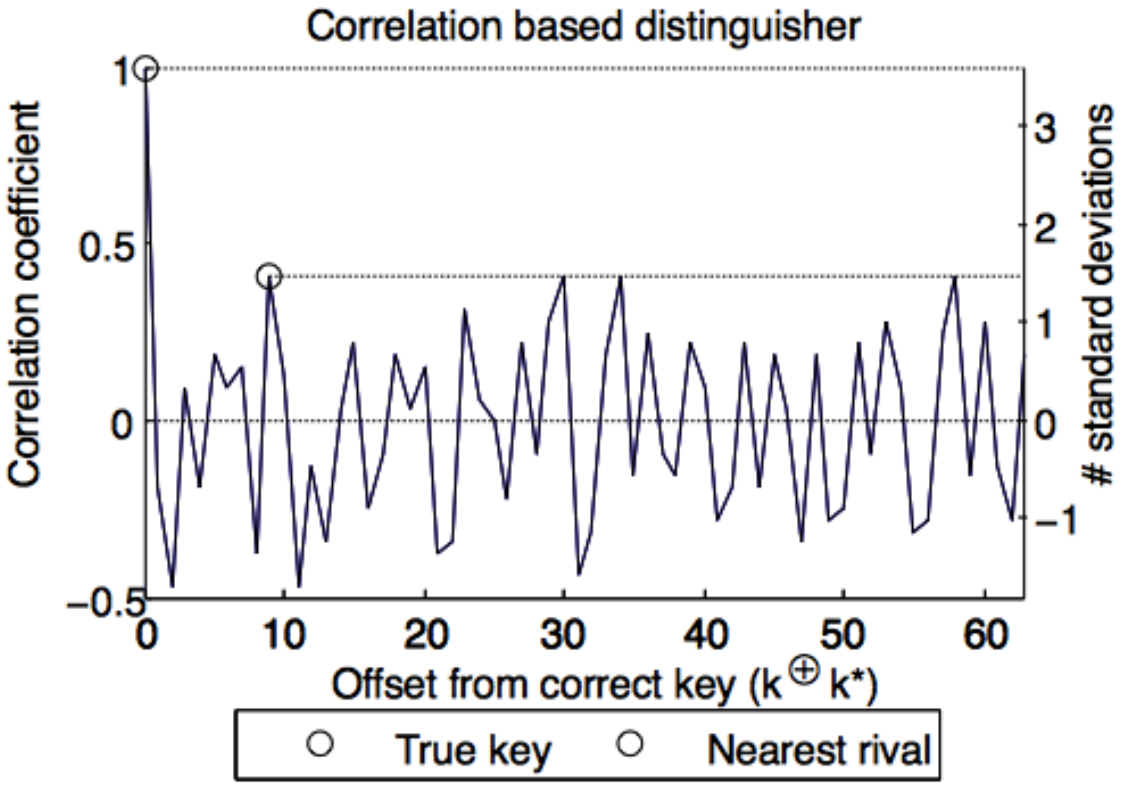
Distinguishers, cont.

- Mangard et al. 2011, 'All for one' paper, IET:
  - Correlation, DoM, and MI are equivalent in 'noisy' enough settings
- Doget et al., 2011, 'Univariate leakage models', JCEN:
  - Can 'translate' one attack, i.e. attacker power model, in 'any' other (PPA, CPA, etc.)
- None of these papers, methods though can be used to analyse MI, nor can they cope with e.g. skewed distributions (noise, data, masks) etc.



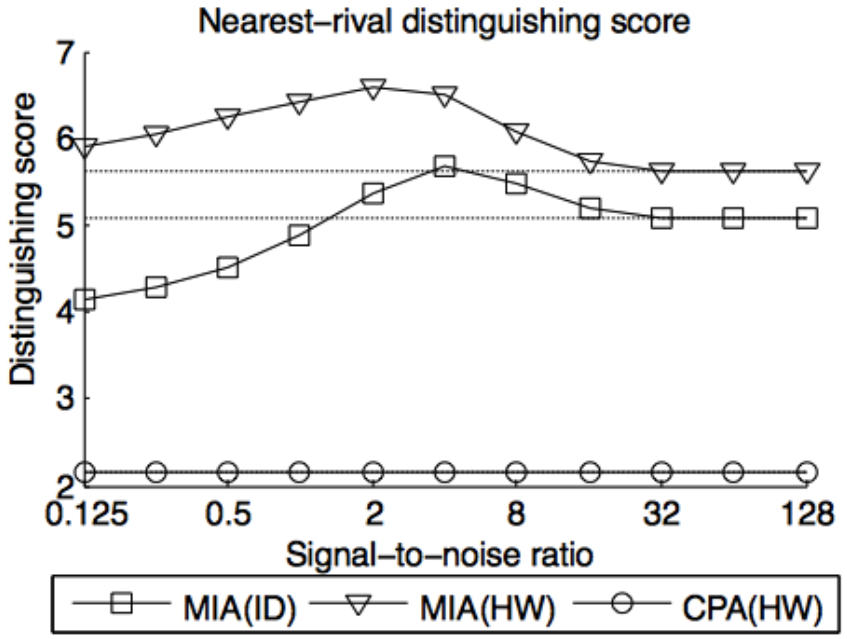
Distinguishers, cont.

- Theoretic distinguishing vectors computationally derived from fully defined distribution
  - No estimation
  - Properties of distinguishing vectors, such as NR margin relevant for practice
  - Whitnall & O., 2011, 'Fair comparison framework …', Crypto 2011



Distinguishers, etc.

- Theoretic DV shows true underlying features:
  - Picture gives evidence that MIA in a noisy scenario reflects is extreme noise sensitivity
  - It can even benefit from noise: stochastic resonance



- Because we are working with theoretic DV we can be sure that this is not a statistical artefact!
- Conclusion: MIA marginally better than CPA
  - Practically CPA will outperform MIA in nearly all contexts

Distinguishers: genericity

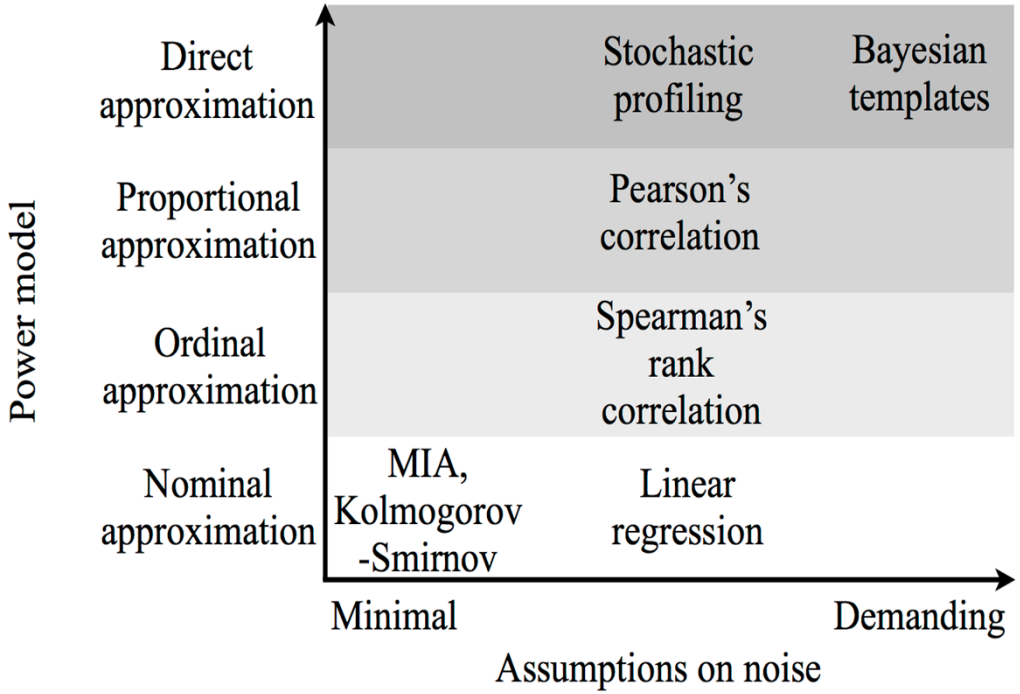
- Another strength of theory is that of having 'formal' definitions
  - Not so much about a scary use of Greek symbols
  - But the attempt to precisely specify what one means
- Example: what is 'generic' DPA?
  - DPA that can be applied in all contexts?
  - DPA that makes no assumptions about the device/leakage model?
  - DPA that uses a generic distinguisher (whatever that might be)?

Distinguishers: genericity, cont.

- No precise definition existed and so different papers approached this differently and discussion mostly centred around 'generic' distinguishers
  - However none of the methods seemed to be able to cope with a bijective S-box unless some power model was supplied …
- Whitnall et al., ePrint, 'Myth & Magic, 2012:
  - Genericity is a property of the power model

Distinguishers: genericity, cont.

- A generic power model is a nominal approximation of the leakage function
  - That is intermediate values are used to separate the true leakage
- A generic distinguisher then is a statistic that can cope with such a nominal model

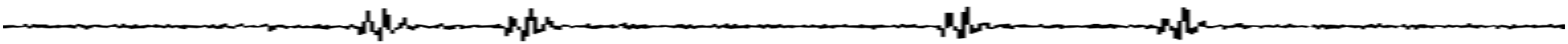


Distinguishers: genericity, cont.

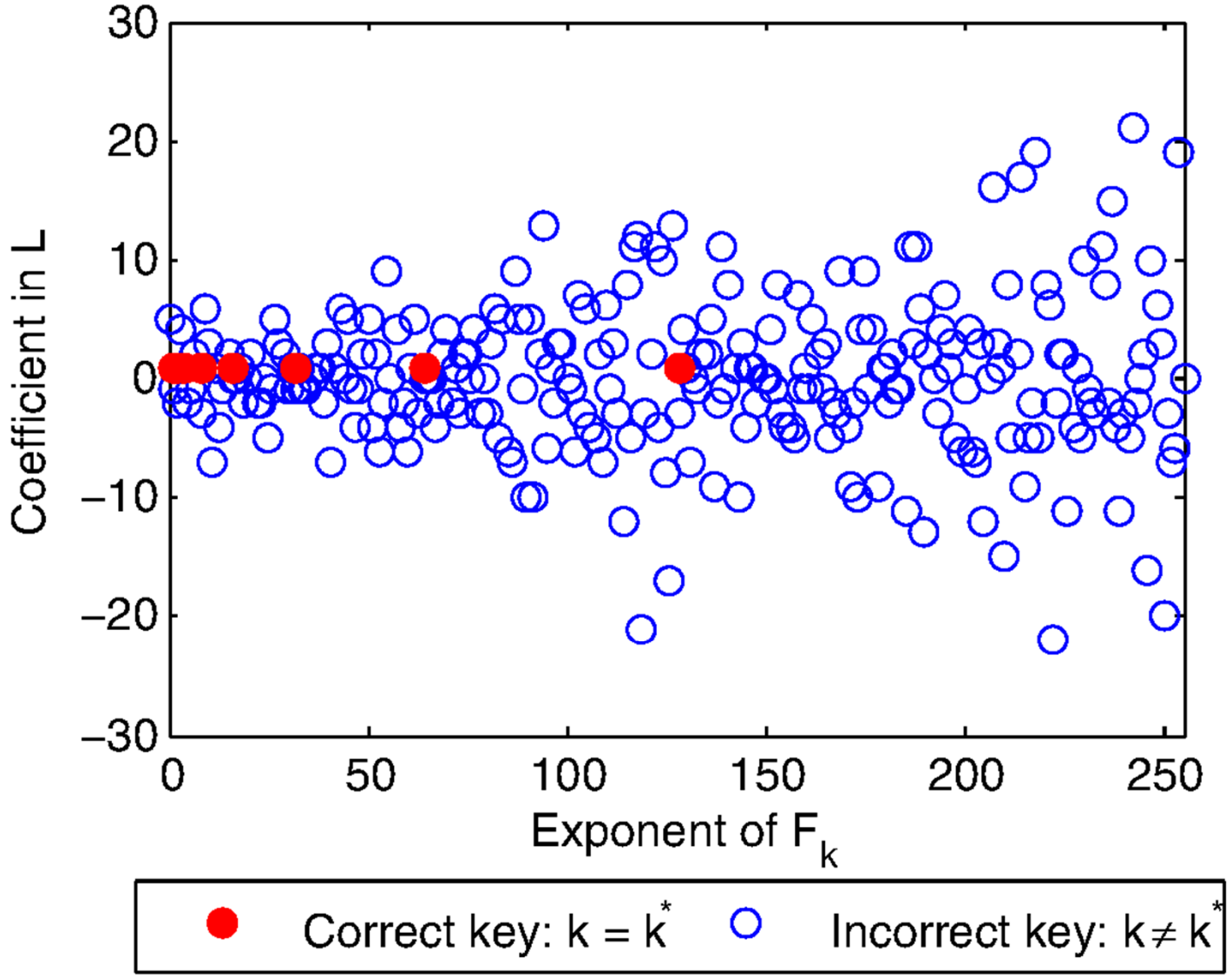
- Now we have a 'clean' definition, so WHAT?
- Nominal models essentially 'classify' power value, this links to classification theory
  - Provides notions to study properties of nominal power models:
    - Precision: prob. that items grouped together belong together
    - Recall: prob. that items that belong together are identified as such
- One can show that any injective composition of target and leakage function gives perfect recall so an attack cannot succeed
  - A further search for distinguishers is hence futile

Distinguishers: genericity, cont.

- Clearly to succeed with a nominal power model in any attack on injective targets some form of 'additional knowledge' is necessary
  - But does it need to be device specific?
- Linear regression based attacks
  - Recover key AND deliver power model
  - Ever looked at those power models?
    - The models for the incorrect key hypotheses (even for simple leakage functions) look 'weird' (unsurprisingly)
- If only one could identify the key via one's 'gut feeling' about the power models ...



Distinguishers: genericity, cont.

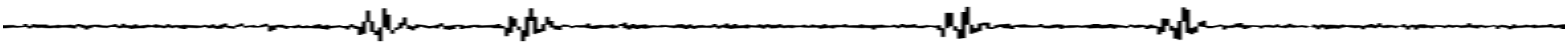


Distiniguisher: genericity, cont.

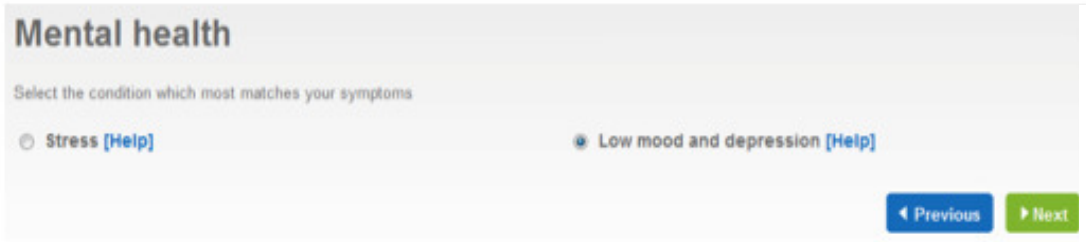
- Stepwise LR: a model building tool used in the machine learning community
  - 'Weird' power models: have lots of terms with little contribution/explanatory power
  - Stepwise LR will (when appropriately configured) will actually omit these terms from the models
    - This means also that their contributions to the R^2 vector which distinguishes the key hypotheses disappears and so key candidates get different R^2. Hence an attack can succeed
- We call this 'generic emulating‘
  - Whitnall et al. ‘Towards DPA attacks without device specific assumptions’ (aka the ‘myth and magic paper’, which can be found on ePrint)

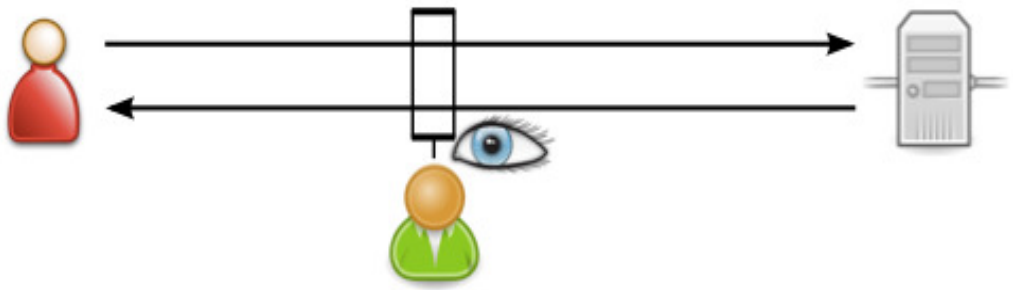
Distinguishers: genericity, cont.

- An aside: stepwise LR is a kind of 'middle ground' with regards to profiling
  - One can use it on unknown devices to recover a 'good enough' power model for DPA key recovery
  - Number of traces, and quality of recovered power model indeed lies well between the optimal DPA (using the true power model) and the most expensive templating methods
- Whitnall & O.: 'Profiled DPA: Efficacy and Efficiency Trade-offs', in submission



Leakage detection: web apps





1 - Stress

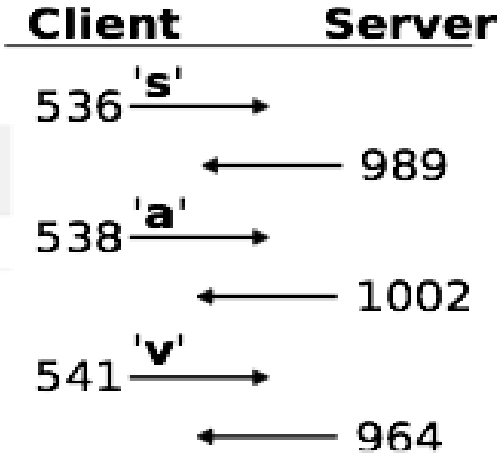
67	12.757846	94.236.79.21	192.168.0.3	TLSv1	1434 Ignored Unknown Reco
68	12.757878	192.168.0.3	94.236.79.21	TCP	66 49861 > https [ACK]
69	12.758413	94.236.79.21	192.168.0.3	TLSv1	1434 Ignored Unknown Reco
70	12.758428	192.168.0.3	94.236.79.21	TCP	66 49861 > https [ACK]
71	12.758445	94.236.79.21	192.168.0.3	TLSv1	1434 Ignored Unknown Reco
72	12.758451	192.168.0.3	94.236.79.21	TCP	66 49861 > https [ACK]
73	12.759992	94.236.79.21	192.168.0.3	TLSv1	1434 Ignored Unknown Reco

2 - Low mood..

114	43.789872	192.168.0.3	94.236.79.21	TCP	66 49861 > https [ACK]
115	43.791206	192.168.0.3	94.236.79.21	TLSv1	803 Application Data
116	43.816221	94.236.79.21	192.168.0.3	TLSv1	970 Application Data
117	43.816255	192.168.0.3	94.236.79.21	TCP	66 49861 > https [ACK]
118	43.873868	94.236.79.21	192.168.0.3	TLSv1	429 Application Data
119	43.873907	192.168.0.3	94.236.79.21	TCP	66 49864 > https [ACK]
120	44.109020	192.168.0.3	94.236.79.21	TLSv1	802 Application Data
121	44.109528	94.236.79.21	192.168.0.3	TLSv1	475 Application Data

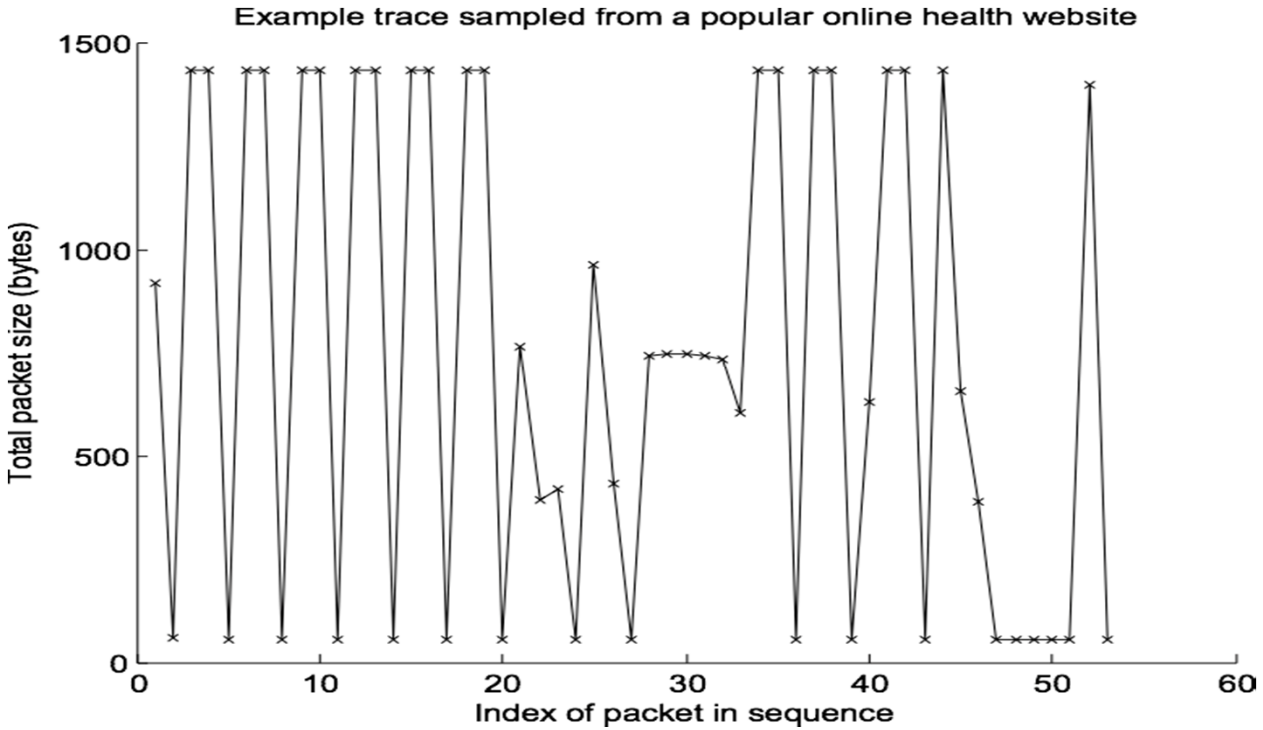
Time Direction Size

Search autocomplete demonstrated to be vulnerable:



Leakage detection: web apps, cont.

Web applications are part of our every day dealings online: health services, tax applications, search applications, etc. The interaction between client and server is leaky.



Leakage detection, cont.

- Leakage detection: spot any point which 'appears' to have non-zero information
- Leakage estimation: take such a point and try and compute or estimate precisely how 'big' this leak is
- Leakage exploitation: take such a point, and use it in some sort of attack

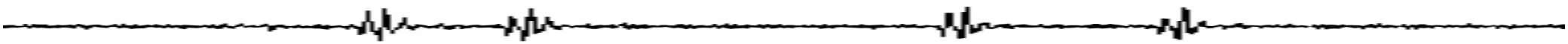
We cannot 'map' easily from one idea to another: detection requires to be less precise than estimation, estimated value does not give any immediate clue with regards to exploitation.

Leakage detection, cont.

- Leak detection was an open issue for Web applications
  - Numerous papers proposed different methods without any statistical rigour
  - MI seemed to be the 'logical' metric
- However only very recently some rigorous statistical tests for MI were developed:
  - Chatzikokolakis et al. (Statistical Measurement of Information Leakage, TACAS 2010) and then
  - Chothia & Guha (A Statistical Test for Information Leaks Using Continuous MI, CSF 2011)

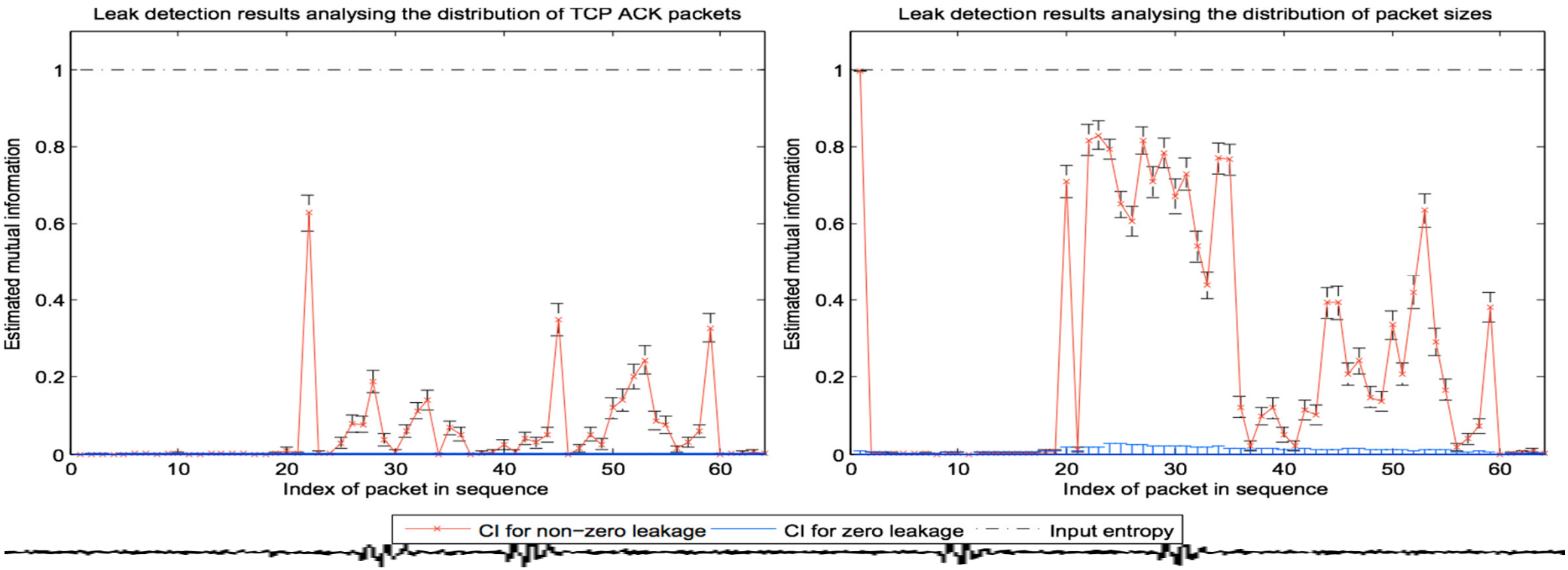
Leakage detection, cont.

- So leakage detection is not just about computing some MI values along a trace
  - It's about determining which values do indicate a non-zero MI in the underlying distribution
  - Need to appropriate statistical tools for our data
  - Using sound statistical technique also allows to integrate prior distributions: very relevant for web application data
- Discrete vs. continuous data



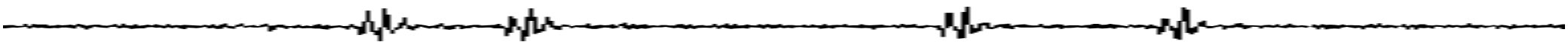
Leakage detection, cont.

- Web applications: leak via different packet-related channels
  - The main challenge is to actually detect ALL the leaks

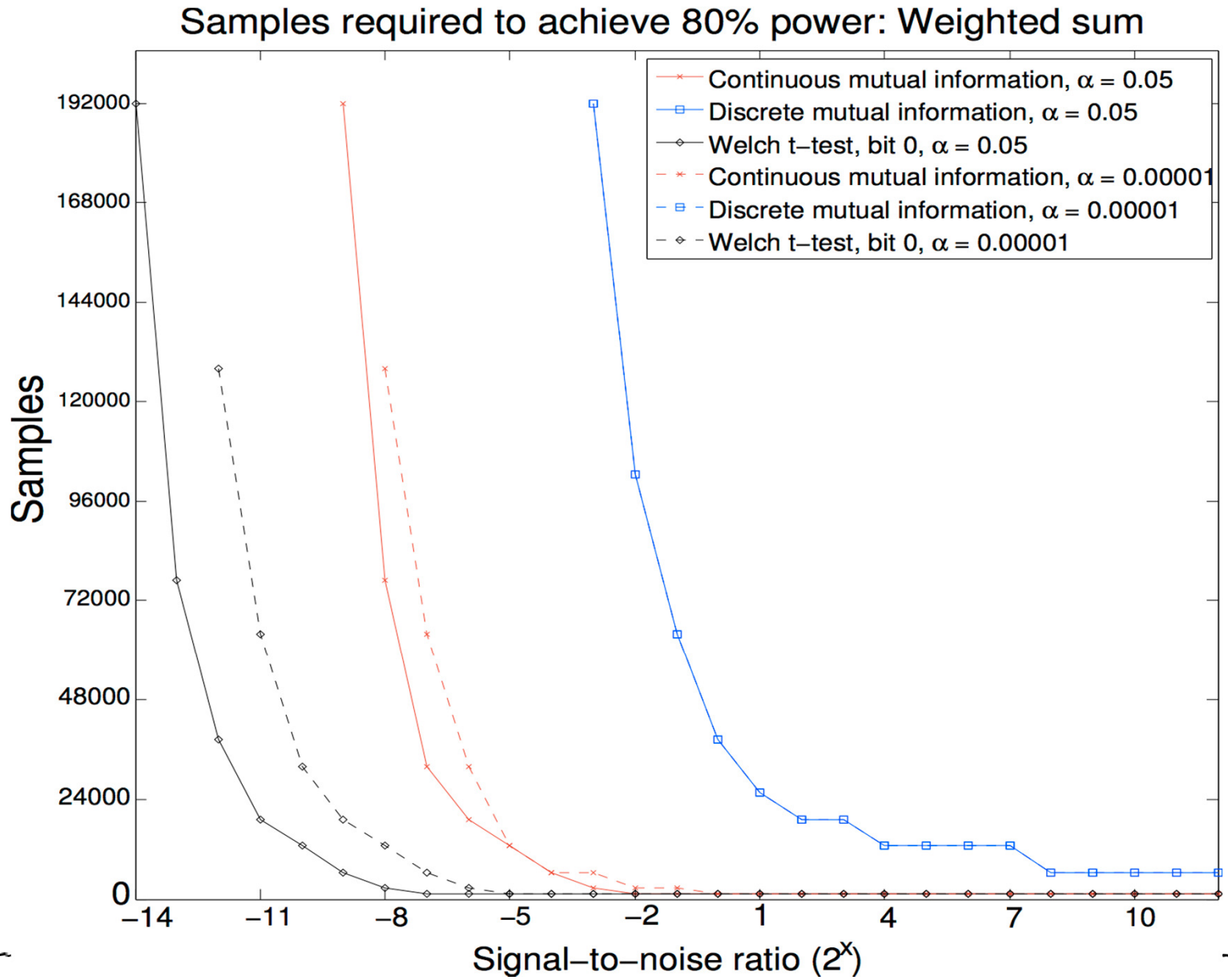


Leakage detection, cont.

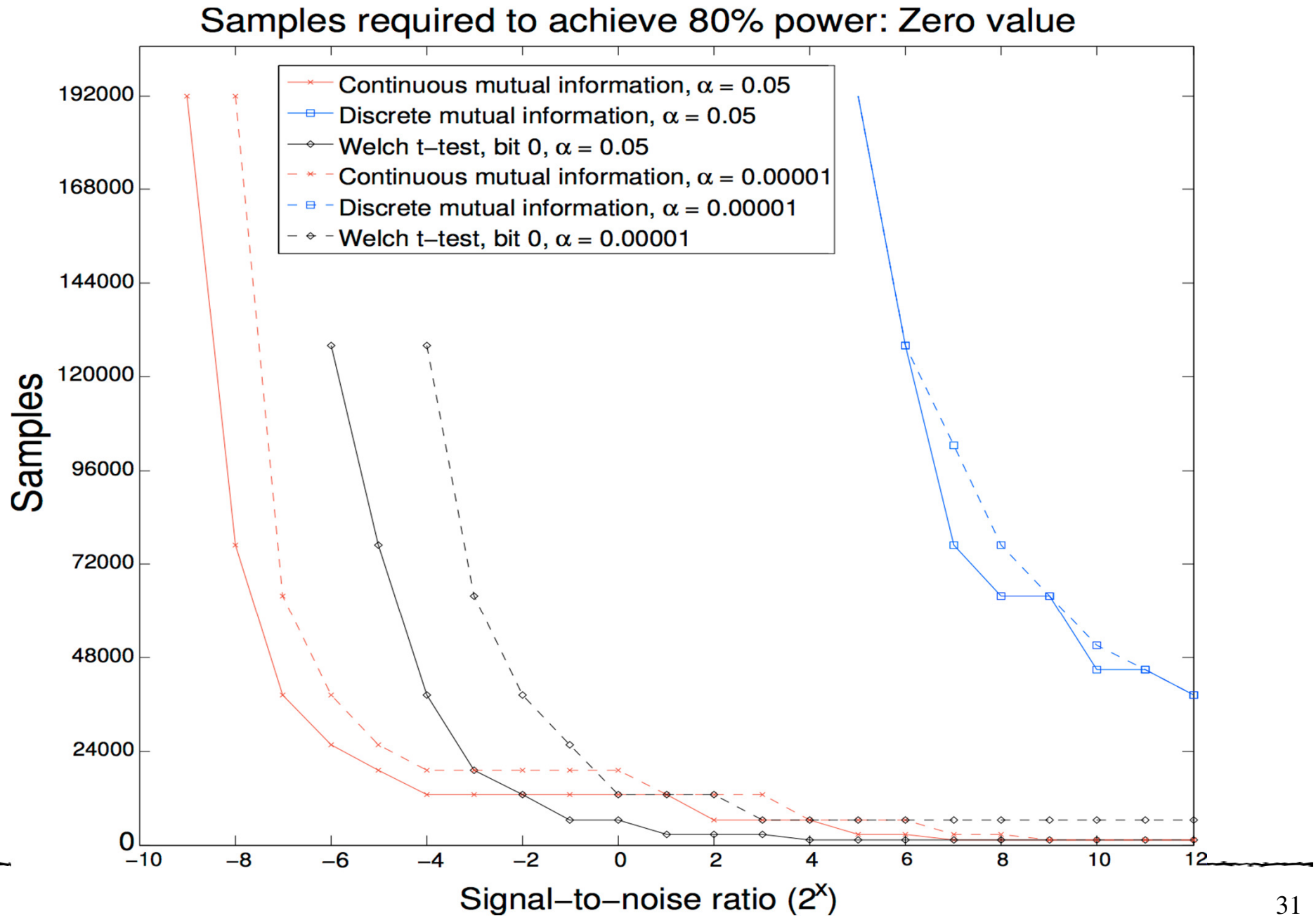
- The idea of using theoretic distinguishing vectors can be spun further to doing an a priori analysis of leakage detection methods also in the case of detection of power/EM leaks
- Recently we studied different leakage detection methods that are used in evaluation of devices against DPA
  - DoM variant: as used by CRI
  - MI tests: as appropriated by us
  - (there is virtually no research on leakage detection)



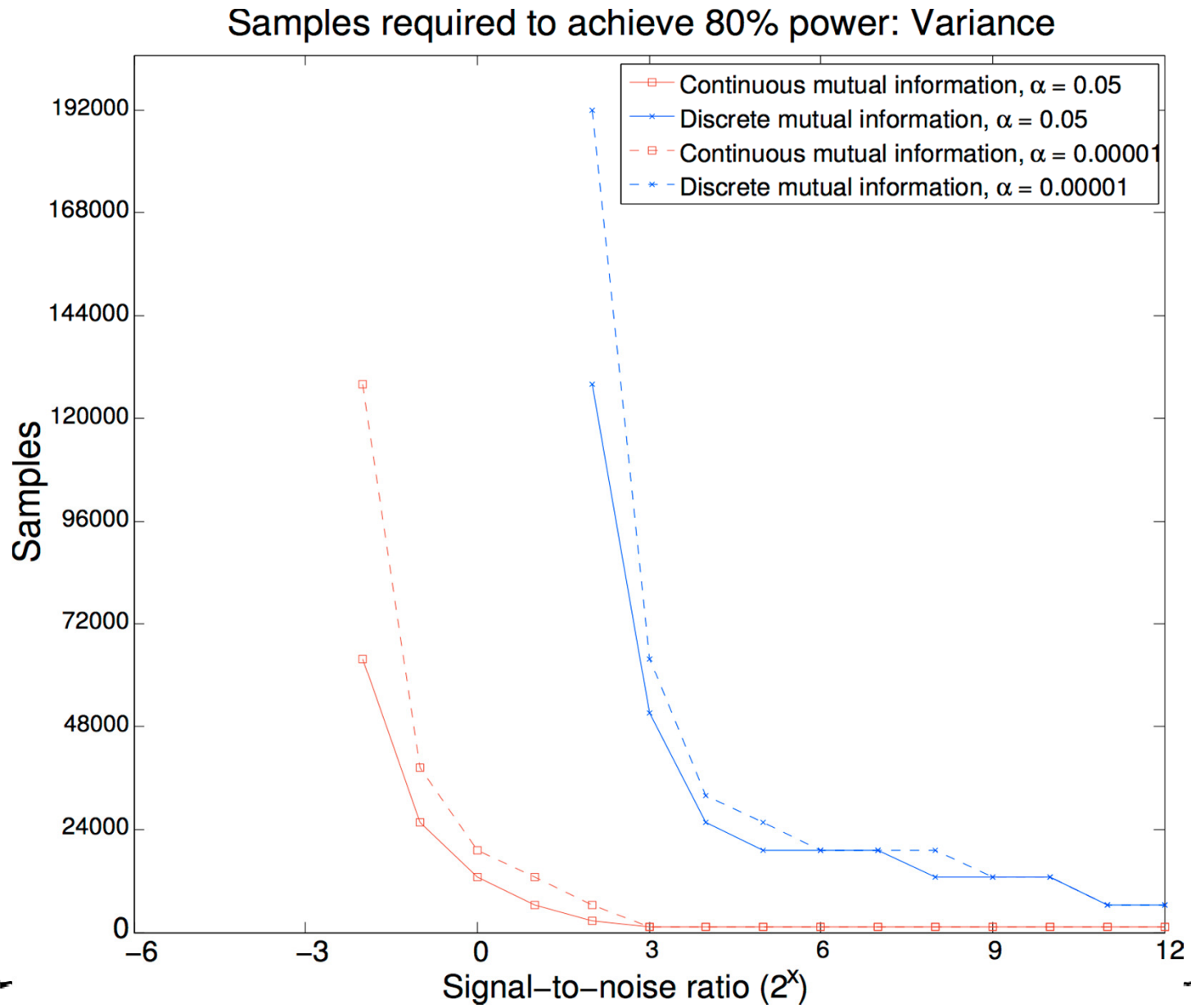
Leakage detection, cont.



Leakage detection, cont.

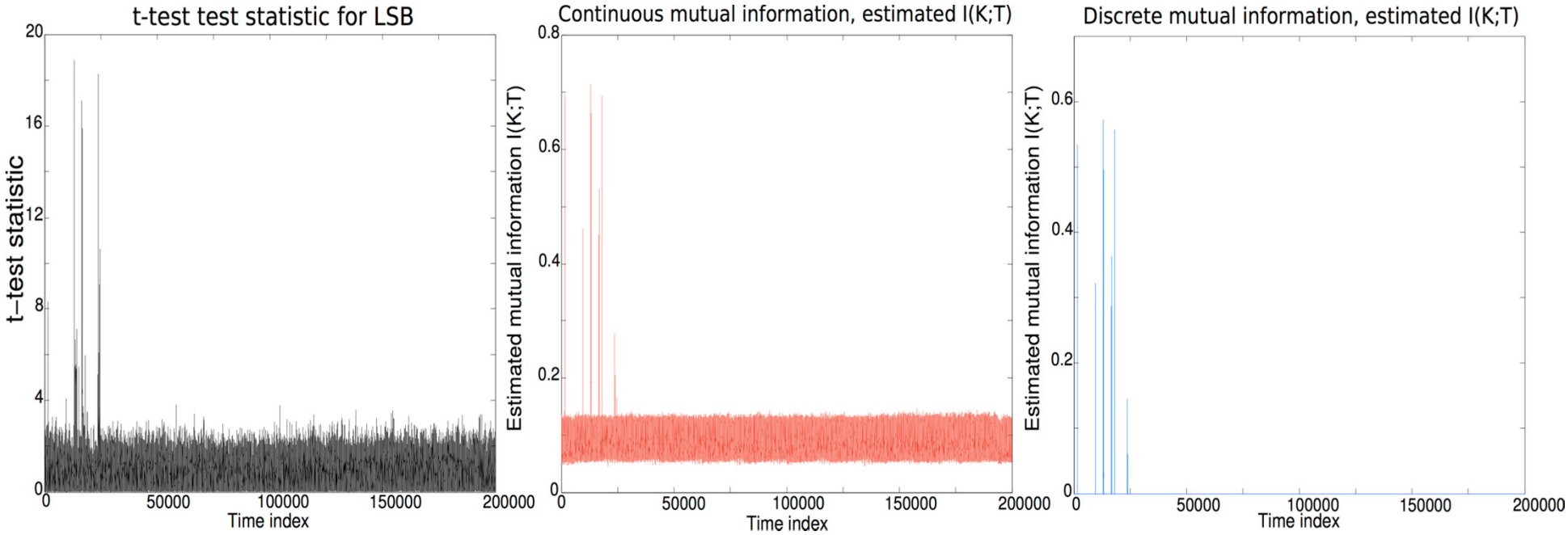


Leakage detection, cont.



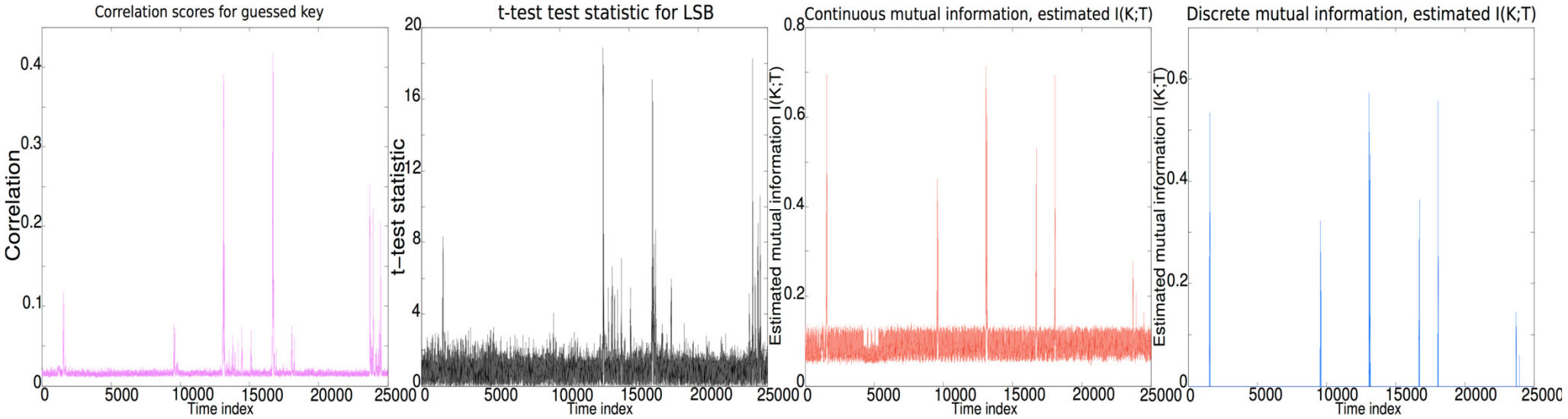
Leakage detection, cont.

- Lot's of points show up with non-zero leakage
- Need a practical way of 'dealing with them'



Leakage detection vs. exploitation

- MI values really don't need to match up with DPA attack results ...



- Mather & Oswald: A Comparison of Statistical Techniques for Detecting Side-Channel Information Leakage in Cryptographic Devices, in submission

Time to conclude

- 'Theory' comes in many forms. To improve SCA research we need:
  - Greater precision in formulating the problems we study
  - Greater emphasis in the distinction between evaluating statistical methods as compared to evaluating devices
- With increased precision and the use of theoretic studies we can in fact
  - Compare conclusively the performance of distinguishers, leakage detection methods, basic working of countermeasures, etc.
  - Specify terms such as 'generic' and disprove the mythical existence of such method