# On the Vulnerability of Low Entropy Masking Schemes

Xin Ye, Thomas Eisenbarth

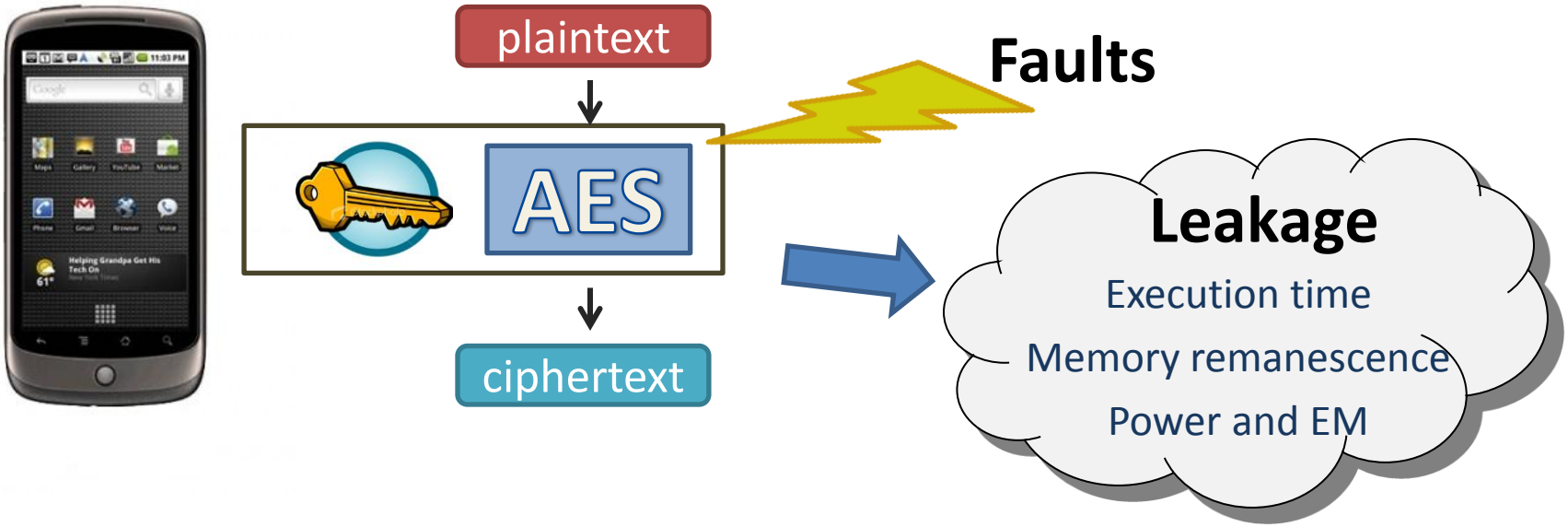
CARDIS 2013 – 11/27/2013


WPI

# Outline

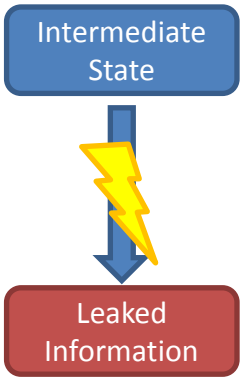
- Masking and Low Entropy Masking (LEMS)
- Ways to exploit remaining leakage
- Collision Attacks on LEMS
- Results on DPA contest v4 traces

# Implementation Attacks



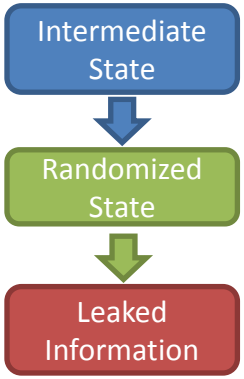

- Critical information leaked through side channels
- Adversary can extract critical secrets (keys etc.)
- Usually require physical access (proximity)

# Ways to Prevent Power Analysis




**Hiding: Decreasing Signal to Noise ratio**
Noise generator, randomized execution order, dual-rail/asynchronous logic styles…

**Problem:** some signal remains, resynchronization, etc.




**Masking: Randomized internal states**
additive/multiplicative masks, Higher-order masking

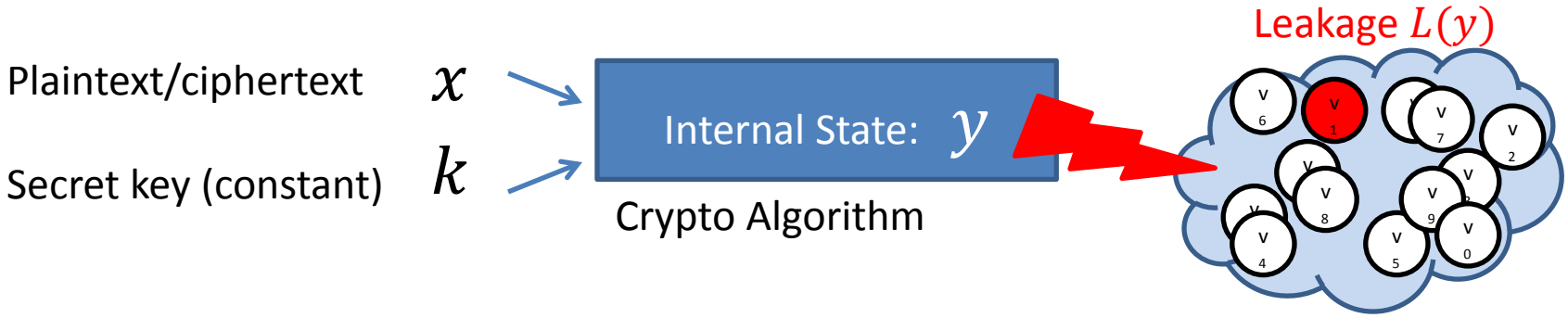
**Problem:** leakage remains, masks also leak

**Effective methods are costly!**

Every single countermeasure can be overcome.

# Masking (concept)

## No Masking:




## Masking:


**Mask ensures that all internal states are equally likely**

# Low Entropy Masking Schemes

**Goal:** Lower implementation cost at *comparable* security:

- no 1st order leakage:
- Resistance against DPA/CPA

- Masks $m$ are from *a subset* of $\{0,1\}^n$
  →low entropy masks

- Self-Complementary Property for masks: $\mathcal{M} = \overline{\mathcal{M}}$
  - For leaking $y_m$, there is a $y_{m'} = \overline{y_m}$ , i.e. bitwise inverse also possible

  →the average leakage is constant

**Claim:** $I(y_m, y) \approx 0$ →negligible mutual information

This is true if uniform input distribution is assumed

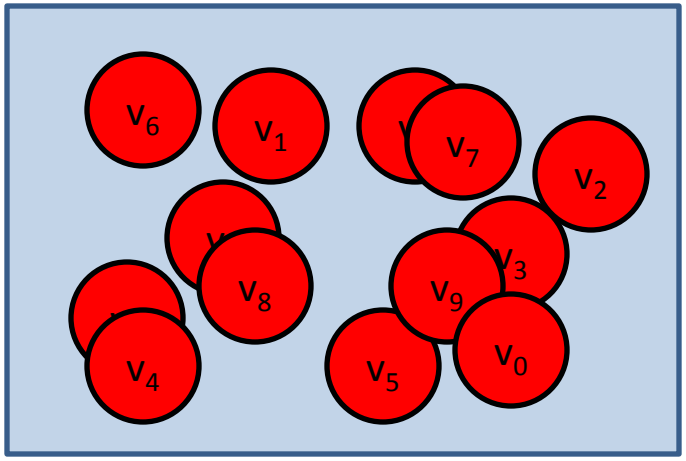
# If we fix an input $x$:

**Classic Masking:** all intermediate values appear with equal probability

**LEMS**: Only few intermediate values possible

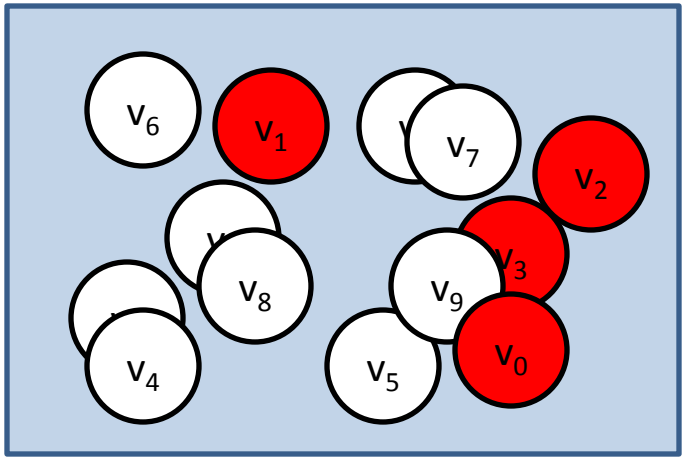
**Full Entropy Masking Schemes (FEMS)**

Fix input x, all values are possible to leak

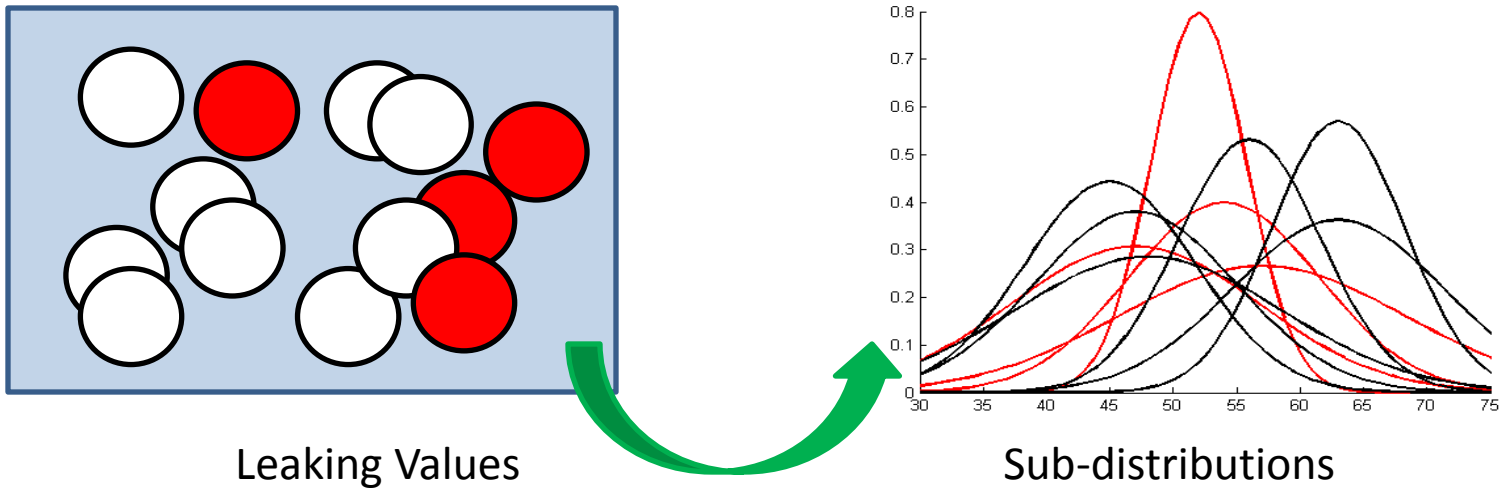



**Low Entropy Masking Schemes (LEMS)**

Fix input x, only a few leaking values



# Leakage Distribution




- Observed distribution for ***fixed input*** is mixture (sum) of leakage of possible masked values

**→ Distributions for *different inputs $x$* are distinguishable**

# Outline

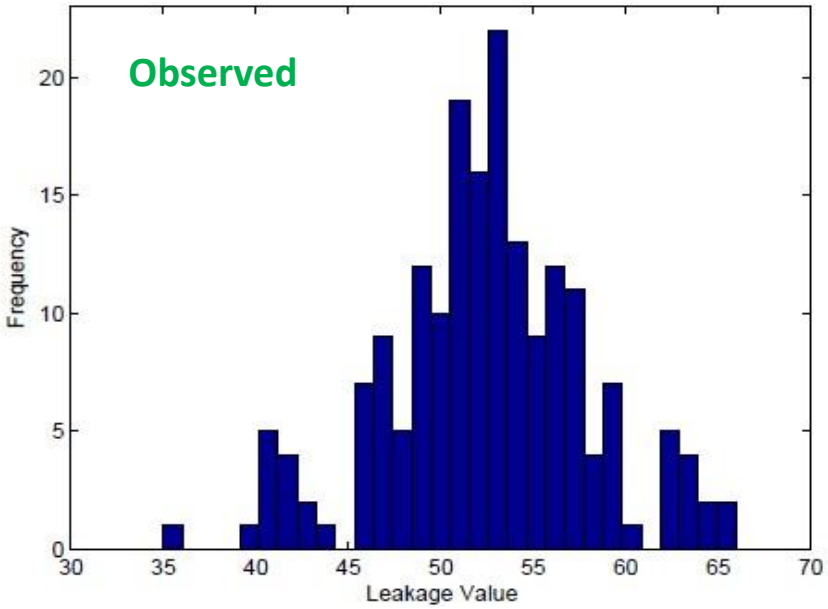
- Masking and Low Entropy Masking (LEMS)
- **Ways to exploit remaining leakage**
- Collision Attacks on LEMS
- Results on DPA contest v4 traces

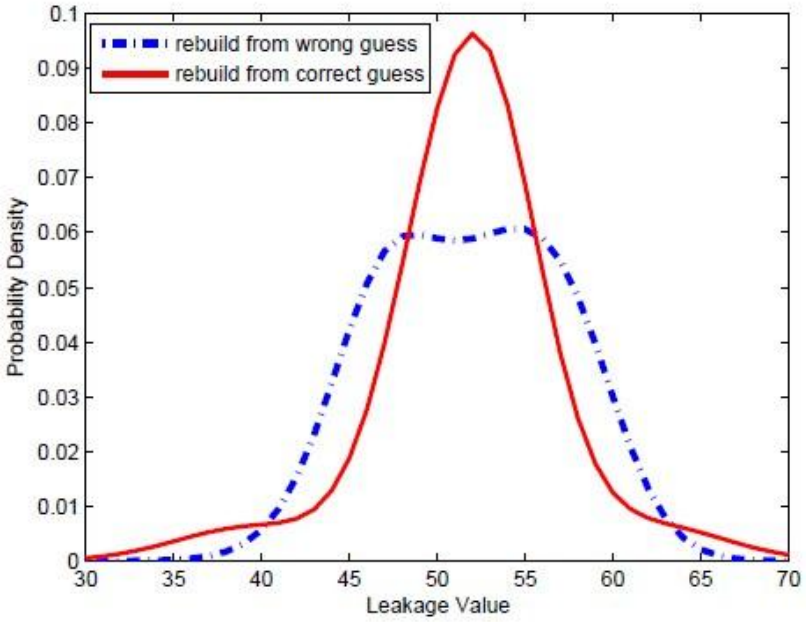
# Leakage Distribution Decomposition Attack

**Concept:** How to test subkey hypothesis:

1. Fix input $x$ and predict leaking set $(\hat{y})_{\mathcal{M}}$
2. Get sub-distributions and rebuild mixture
   → output is leakage distribution for $y_{\mathcal{M}}$
3. Measure closeness between observed and rebuilt distributions

Repeat for all inputs $x$ and all subkey hypotheses $g$



# LDDA: Practicality

**Problem:** How to estimate sub-distributions?

**LDDA with profiling:**

- Assumes known masks during profiling
- Similar to template-like attacks on masking: [SLP05,OM07,LP07]
- **Difference:** Univariate leakage sufficient!

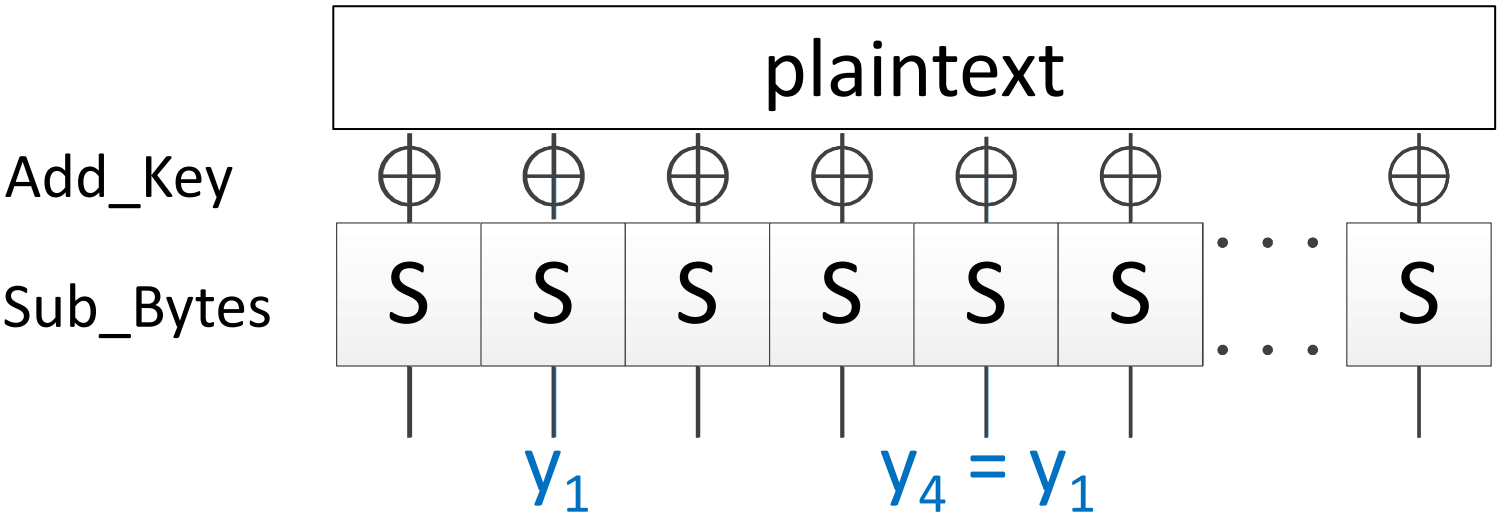
**LDDA without profiling:**

- Assumes linear leakage model, e.g. Hamming weight (similar to linear regression methods)
- Works with unknown masks
- Again, univariate leakage sufficient!

# Outline

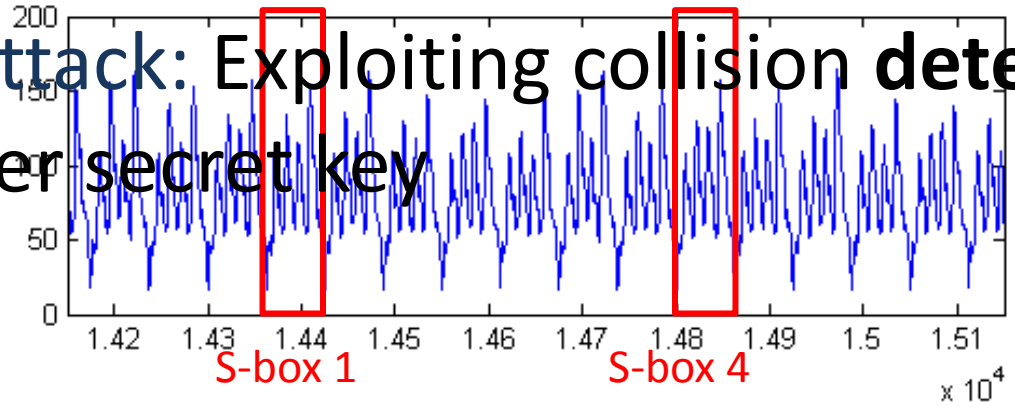
- Masking and Low Entropy Masking (LEMS)
- Ways to exploit remaining leakage
- **Collision Attacks on LEMS**
- Results on DPA contest v4 traces

# Side Channel Collisions in AES




Collision: Querying same S-box value twice

Collision Attack: Exploiting collision **detections** to recover secret key



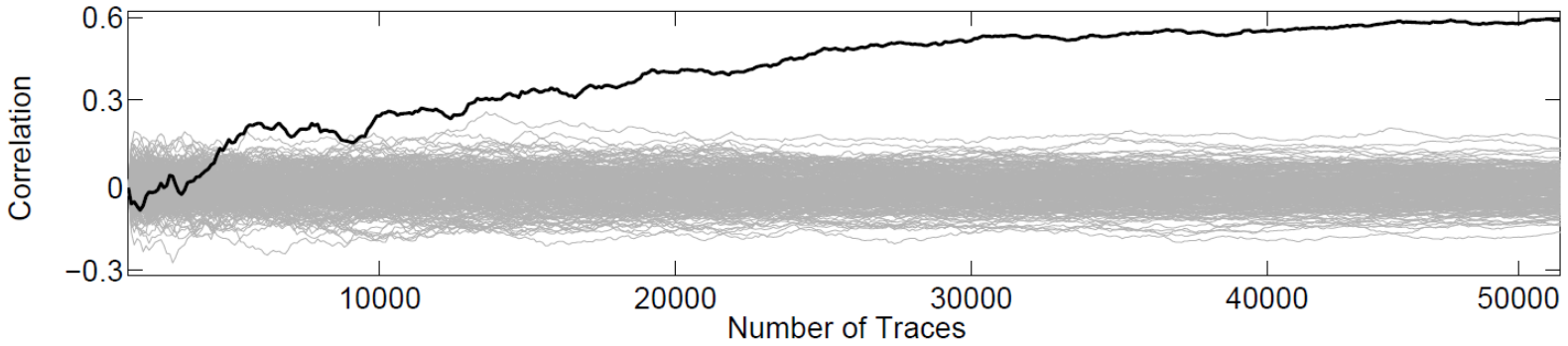
# How to Improve Collisions

Collisions: Simple approach, but requires strong leakage

Improvement: Correlation Collision Attack [MME10]

- Use many measurements
- Compute average for each possible output
- Use **all** S-box output leakages for comparison

→Strong attack, breaks many protected implementations

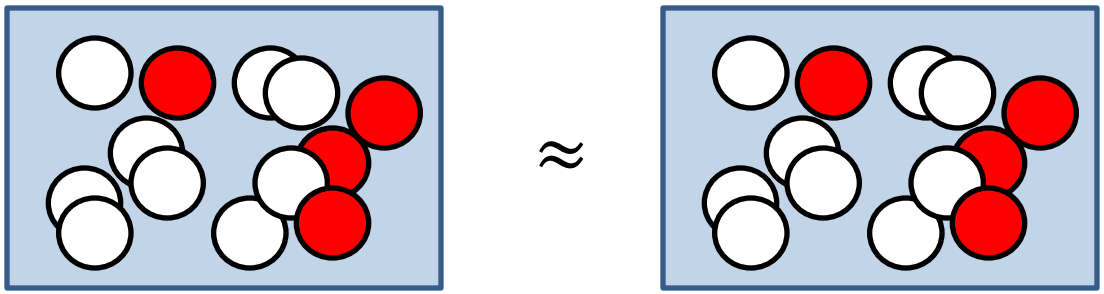


# Leaking Set Collision Attack

- Find two *different* inputs $x \neq x'$ for which the leaking set $(y)_{\mathcal{M}}$ is **identical**
- Exists due to *self-complementary* masks $(m, \bar{m})$

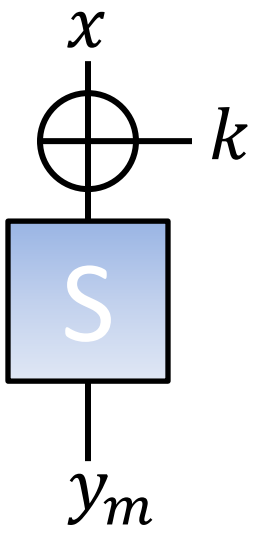
e.g. AES s-box output:

$$\underbrace{S(x \oplus k) \oplus m}_{y_{\mathcal{M}}} = \underbrace{S(x' \oplus k) \oplus \bar{m}}_{y'_{\mathcal{M}}}$$



$\approx$





# Leaking Set Collision Attack (II)

- For **correct** key guess: $y' = \bar{y}$

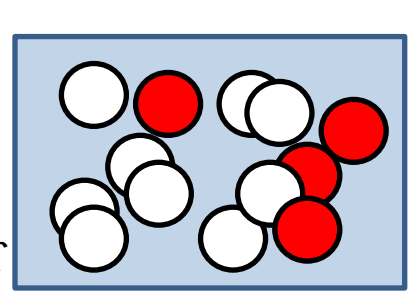
**Leaking Set Collision**

$\Rightarrow (y)_{\mathcal{M}} = (y')_{\mathcal{M}}$

Fix input x

Find $x'$

$x' = f(x, k)$



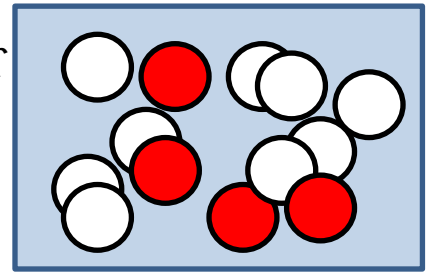

**The Same**

Composition of Sub-distributions

- For **wrong** key guess: $y' \neq \bar{y}$

$x' = f(x, g \neq k)$

$\Rightarrow (y)_{\mathcal{M}} \neq (y')_{\mathcal{M}}$




**Different**

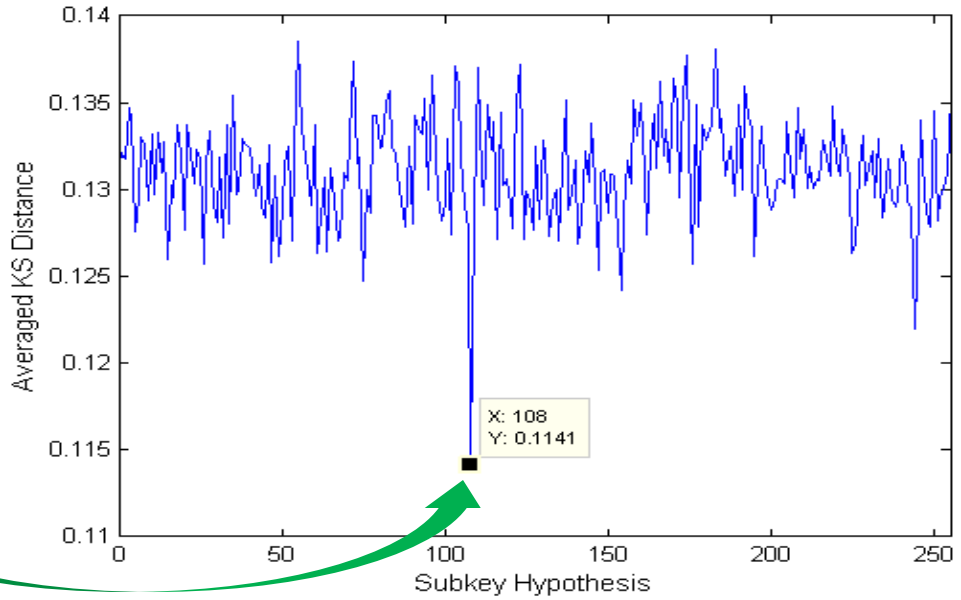
Composition of Sub-distributions

- Distance Metric: Kolmogorov-Smirnov (KS-distance)

# Leaking Set Collision Attack (III)

1. Derive set collisions for masked AES Sbox output
   $$x' = f(x,k) = k \oplus S^{-1}(\text{0xff} \oplus S(x \oplus k))$$
1. Compare observed leakage distributions
2. Choose key guess with lowest distance

**Correct Hypothesis**
**➔ Low Distance**



# Leaking Set Collision Attack (IV)

- Like Correlation Collision Attack, all traces are grouped and compared
- Unlike correlation collision attacks, works on inputs for the *same* s-box
  (same univariate leakage point)
- Needs sufficient measurements to approximate distribution

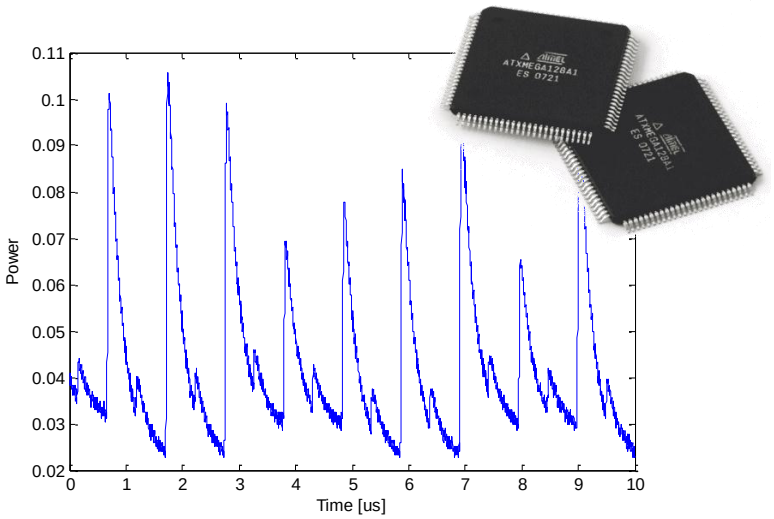
# Outline

- Masking and Low Entropy Masking (LEMS)
- Ways to exploit remaining leakage
- Collision Attacks on LEMS
- **Results on DPA contest v4 traces**

# Experimental Results

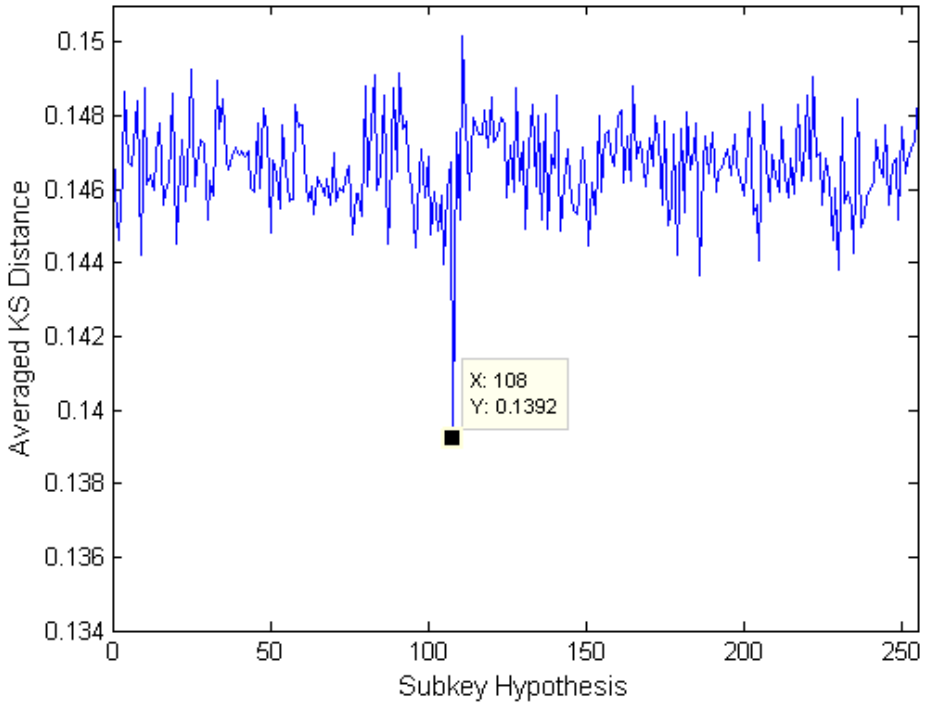
**Target**: RSM AES-256 software implementation from DPA contest v4:

- 8-bit microcontroller (strong leakage)
- 16 self-complementary masks
- 100.000 traces available (known mask and key)
- Attack performed on s-box output



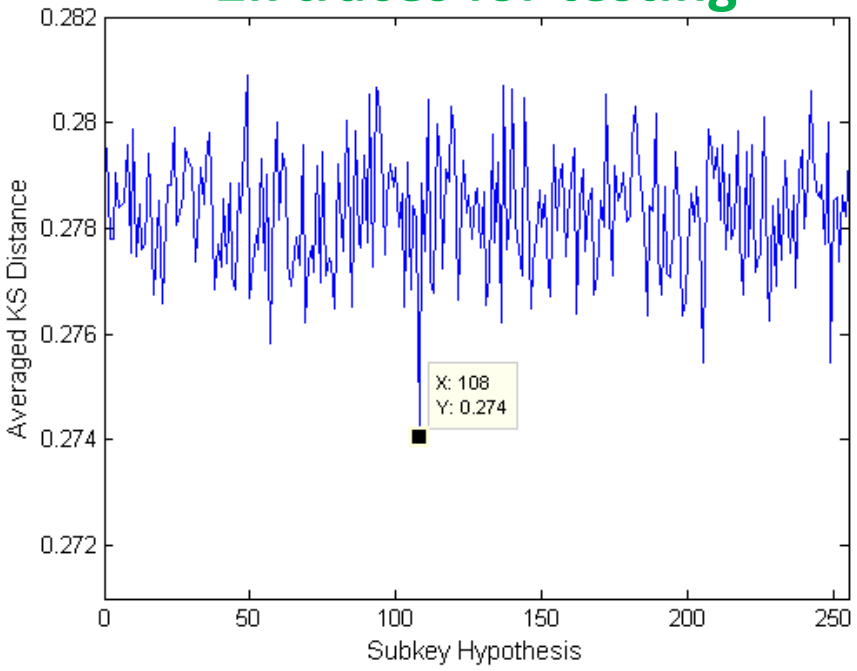
# LDDA with profiling

- 50,000 traces to build univariate templates (i.e. sub-distributions)
- 8k traces to test subkey hypotheses (2k, 16k next slide)
- Mask known during profiling
- KS-distance (y-axis) to measure similarity

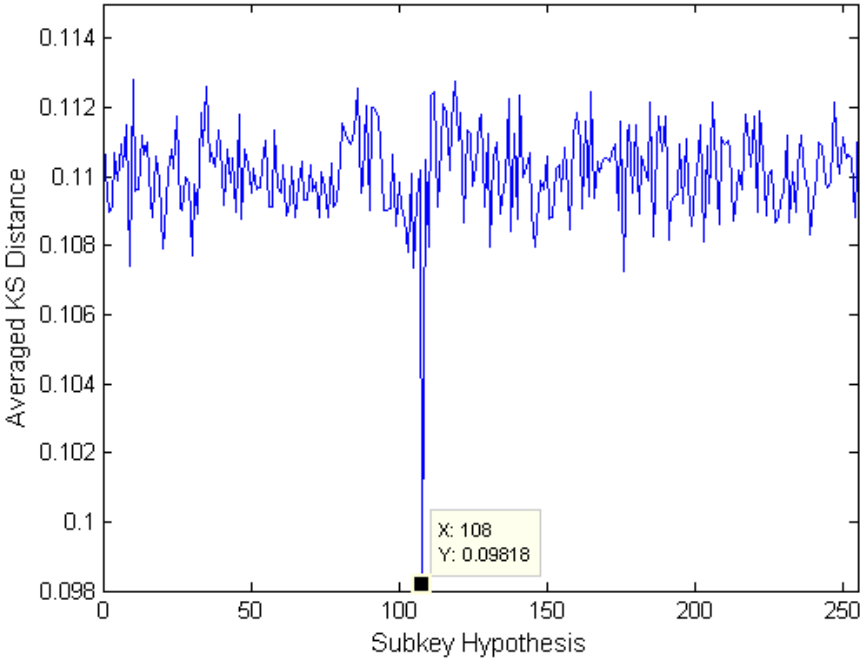


# LDDA with profiling

**2k traces for testing**




**16k traces**




- Overall distance decreases
- Correct key is better distinguishable with increased number of measurements

# LDDA without profiling

- Assumed leakage model: Hamming Weight
- Parameters estimated over all traces
- Outcome depends on parameter choice

| Number of Traces | 20k | 40k | 60k | 80k | 100k |
|---|---|---|---|---|---|
| GE average case | 19.74 | 16.65 | 4.02 | 2.93 | 1.31 |
| GE worst case | 30 | 33 | 11 | 9 | 5 |
| GE best case | 9 | 2 | 2 | 1 | 1 |

- Attack feasible even with imperfect model

# Leaking Set Collision Attack

| Number of Traces | 16 x 256 | 32 x 256 | 48 x 256 | 64 x 256 |
|---|---|---|---|---|
| Guessing Entropy | 46.78 | 17.78 | 7.00 | 1.00 |
| 1st order Succ Rate | 5.56% | 44.4% | 83.3% | 100.0% |
| 4th order Succ Rate | 33.3% | 55.6% | 83.3% | 100.0% |

→Clear distinguishability with 16k traces

# Conclusions

- Low-entropy masking schemes have distinguishable leakage distributions
- “Efficient” univariate attacks exploiting this leakage are available
- Self-complementary masks enable self-collision attacks: Leaking Set Collision Attacks

# Thank you for your attention!

users.wpi.edu/~teisenbarth

teisenbarth@wpi.edu

xye@wpi.edu

- References

[SLP05] A stochastic model for differential side channel cryptanalysis; W. Schindler, K. Lemke, and C. Paar

[OM07] Template attacks on masking – resistance is futile; E. Oswald and S. Mangard

[LP07] Analyzing side channel leakage of masked implementations with stochastic methods; K. Lemke-Rust and C. Paar

[DPRS11] Univariate side channel attacks and leakage modeling; J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert

[MME10] Correlation-enhanced power analysis collision attack; A. Moradi, O. Mischke, T. Eisenbarth