Efficient Template Attacks

CARDIS 2013

Omar Choudary Markus G. Kuhn

UNIVERSITY OF CAMBRIDGE

Berlin, 29 November 2013

Introduction

- Template Attacks [Chari et al., '03]

Introduction

- Template Attacks [Chari et al., '03]
- Certification to CC profiles requires their evaluation

Introduction

- Template Attacks [Chari et al., '03]
- Certification to CC profiles requires their evaluation
- Contributions:
    - Dealing with large number of samples
      (avoiding numerical pitfalls)

Introduction

- Template Attacks [Chari et al., '03]
- Certification to CC profiles requires their evaluation
- Contributions:
  - Dealing with large number of samples (avoiding numerical pitfalls)
  - Efficient implementation (reducing evaluation time, e.g. from 3 days to 30 minutes)

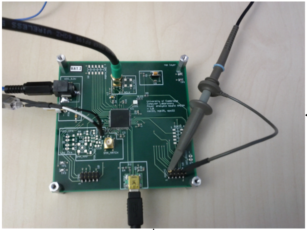
Introduction

- Template Attacks [Chari et al., '03]
- Certification to CC profiles requires their evaluation
- Contributions:
  - Dealing with large number of samples
    (avoiding numerical pitfalls)
  - Efficient implementation
    (reducing evaluation time, e.g. from 3 days to 30 minutes)
  - Fair evaluation of most common compression techniques
    - Show several assumptions do not hold in general
    - Practical guideline for choosing the right compression

Introduction

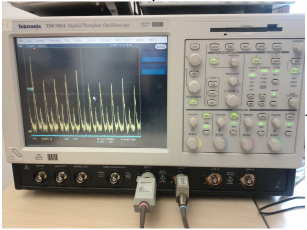
- Template Attacks [Chari et al., '03]
- Certification to CC profiles requires their evaluation
- Contributions:
  - Dealing with large number of samples
    (avoiding numerical pitfalls)
  - Efficient implementation
    (reducing evaluation time, e.g. from 3 days to 30 minutes)
  - Fair evaluation of most common compression techniques
    - Show several assumptions do not hold in general
    - Practical guideline for choosing the right compression
  - And ... we provide data and code so you can try it!

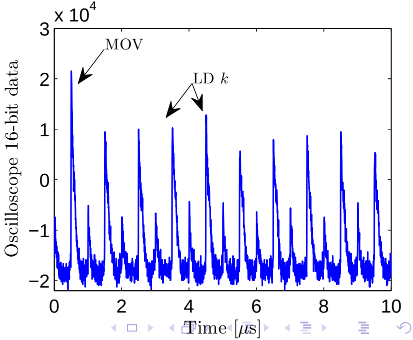
Experiment: eavesdropping on 8-bit data bus



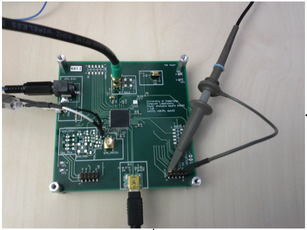
Executed Code:

```
movw r30, r24
ld r8, Z+
ld r9, Z+
ld r10, Z+
ld r11, Z+
```



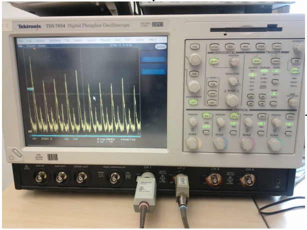


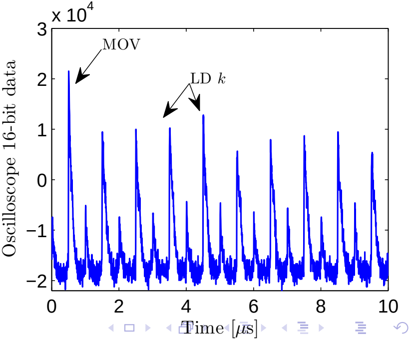
Experiment: eavesdropping on 8-bit data bus



Executed Code:

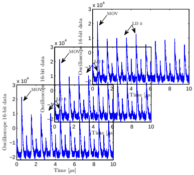
```
movw r30, r24
ld r8, 0
ld r9, k
ld r10, 0
ld r11, 0
```



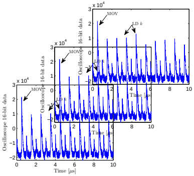


Profiling: Acquire Traces

$k = 0$



$k = 1$



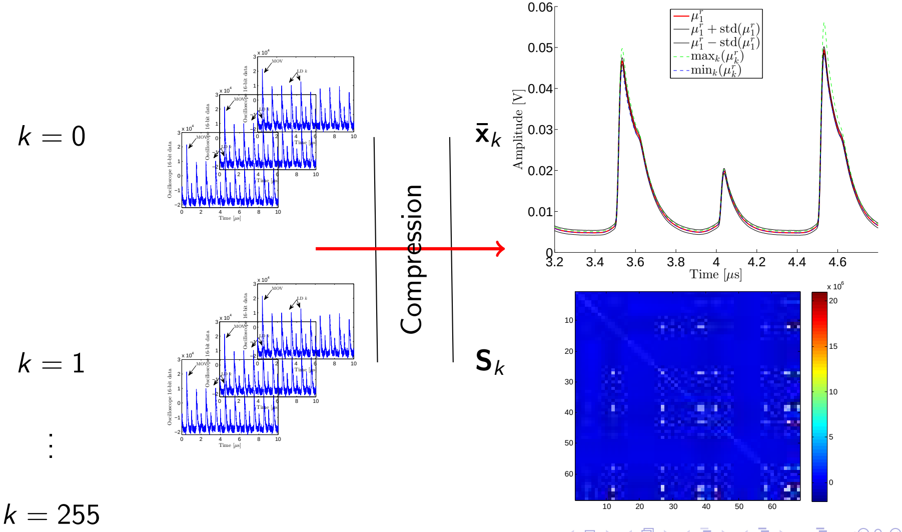
⋮

$k = 255$

Executed Code:

```
movw r30, r24
ld r8, 0
ld r9, k
ld r10, 0
ld r11, 0
```

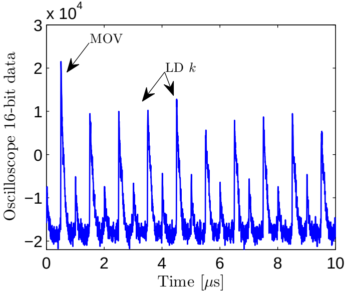
Profiling: Estimate Templates

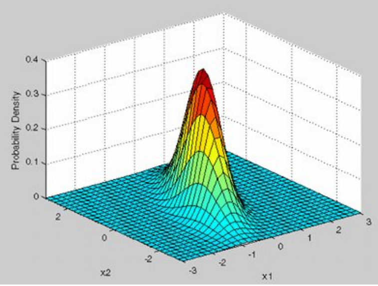


Attack: using the multivariate normal distribution

$$\mathrm{d}(k \mid \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{S}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_k)' \mathbf{S}_k^{-1} (\mathbf{x} - \bar{\mathbf{x}}_k)\right)$$

$k\star \rightarrow \mathrm{argmax}_k\, \mathrm{d}(k \mid \mathbf{x})$





Problem 1: Floating point issues

$$\mathrm{d}(k \mid \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{S}_k|}} \boxed{\exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_k)'\mathbf{S}_k^{-1}(\mathbf{x}-\bar{\mathbf{x}}_k)\right)}$$

- Issue 1: $\exp(x)$ is only safe for $|x| < 710$, which is easily exceeded in our experiments.

Problem 1: Floating point issues

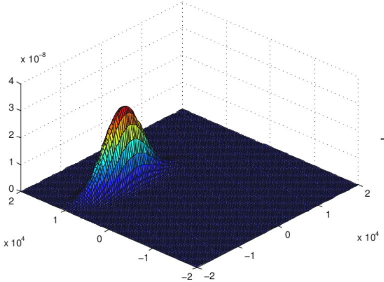
$$\mathrm{d}(k \mid \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m \boxed{|\mathbf{S}_k|}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_k)'\mathbf{S}_k^{-1}(\mathbf{x} - \bar{\mathbf{x}}_k)\right)$$

- Issue 1: $\exp(x)$ is only safe for $|x| < 710$, which is easily exceeded in our experiments.
- Issue 2: $|\mathbf{S}_k|$ can overflow/underflow easily for large m (> 50).

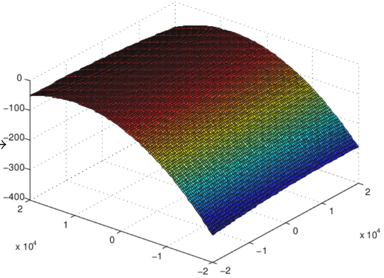
These are *real* problems. Naive implementations are likely to fail.

Solution: use LOG

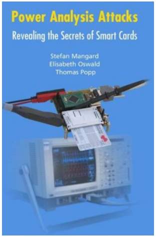
$$\mathrm{d}_{\mathrm{LOG}}(k \mid \mathbf{x}) = -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{S}_k| - \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_k)'\mathbf{S}_k^{-1}(\mathbf{x}-\bar{\mathbf{x}}_k)$$



$\xrightarrow{\log}$

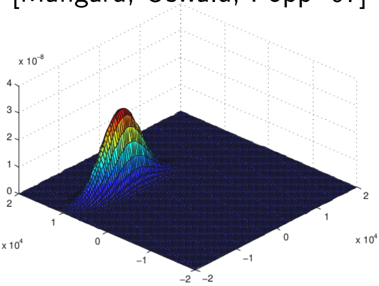


Caveat: pdf can be larger than 1

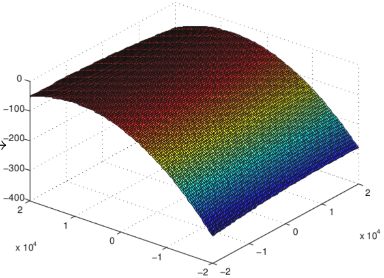


“[Choose the candidate *k* that leads to the] *smallest* absolute value [of $\mathrm{d}_{\mathrm{LOG}}$]”

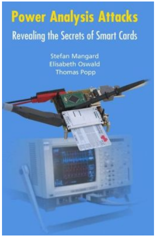
[Mangard, Oswald, Popp ’07]



$\xrightarrow{\log}$



Caveat: pdf can be larger than 1



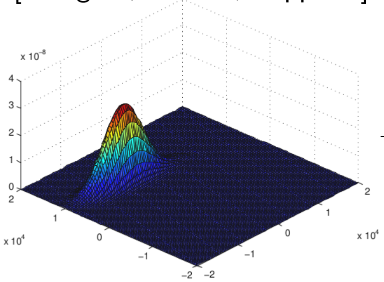
"[Choose the candidate *k* that leads to the] *smallest* absolute value [of $\mathrm{d}_{\mathrm{LOG}}$]"

Incorrect:

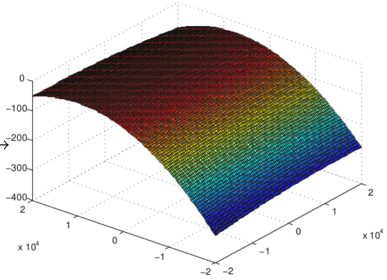
log is monotonic, abs is not!

We choose *k* with *highest* value of $\mathrm{d}_{\mathrm{LOG}}$.

[Mangard, Oswald, Popp '07]



$\xrightarrow{\log}$



Problem 2: dealing with large number of samples

- Myth: problems with inversion of $\mathbf{S}_k$ as soon as m is large.

m = number of samples
n_{p} = number of traces from profiling, for each k

Problem 2: dealing with large number of samples

- Myth: problems with inversion of $\mathbf{S}_k$ as soon as m is large.
- Clarification:
  - $n_\mathrm{p} \leq m$: $\mathbf{S}_k$ cannot be inverted $(\mathrm{rank}(\mathbf{S}_k) < n_\mathrm{p})$

m = number of samples
n_p = number of traces from profiling, for each k

Problem 2: dealing with large number of samples

- Myth: problems with inversion of $\mathbf{S}_k$ as soon as m is large.
- Clarification:
  - $n_{\mathrm{p}} \leq m$: $\mathbf{S}_k$ cannot be inverted $(\mathrm{rank}(\mathbf{S}_k) < n_{\mathrm{p}})$
  - $n_{\mathrm{p}} > m$: $\mathbf{S}_k$ will most likely be invertible (ignoring highly correlated samples)

m = number of samples

n_{p} = number of traces from profiling, for each k

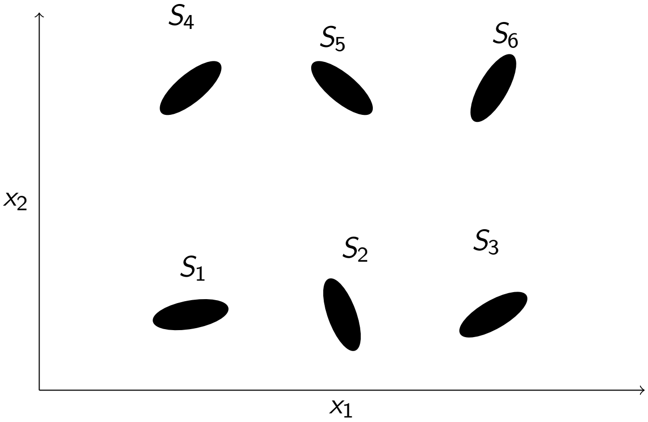
Problem 2: dealing with large number of samples

- Myth: problems with inversion of $\mathbf{S}_k$ as soon as m is large.
- Clarification:
  - $n_\mathrm{p} \leq m$: $\mathbf{S}_k$ cannot be inverted $(\mathrm{rank}(\mathbf{S}_k) < n_\mathrm{p})$
  - $n_\mathrm{p} > m$: $\mathbf{S}_k$ will most likely be invertible (ignoring highly correlated samples)
- Problem: obtaining $n_\mathrm{p} > m$ can be difficult due to memory and time constraints.

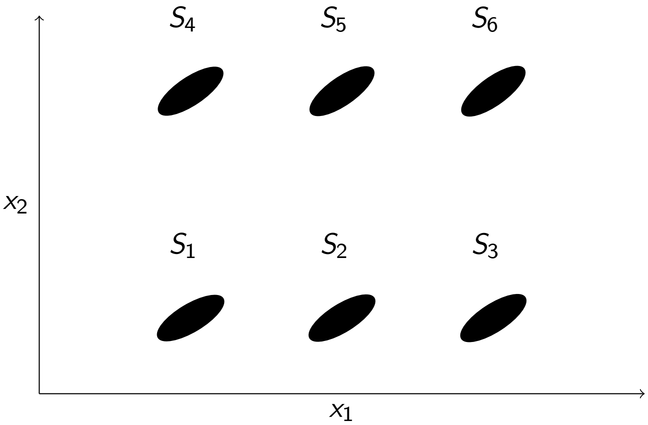
m = number of samples

n_p = number of traces from profiling, for each k

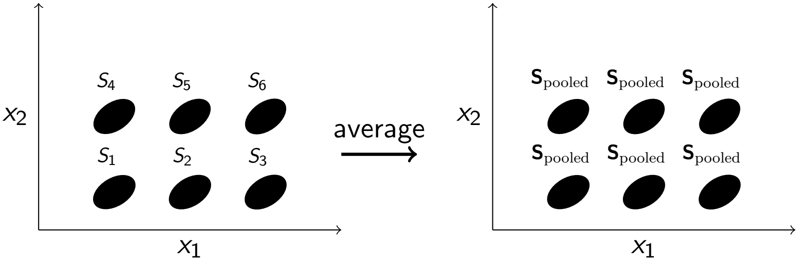
Scenario 1: $\mathbf{S}_k$ dependent on k



Scenario 2: $\mathbf{S}_k$ independent on k



Efficient solution: use $\mathbf{S}_{\mathrm{pooled}}$



- $\mathbf{S}_{\mathrm{pooled}}$ is an average of the covariances.
- $\mathbf{S}_{\mathrm{pooled}}$ uses $|\mathcal{S}|n_{\mathrm{p}}$ traces, while $\mathbf{S}_k$ only n_{p}.
- Now the condition for non-singularity is $n_{\mathrm{p}} > \frac{m}{|\mathcal{S}|}$
    - A great advantage in practice.

Mahalanobis Distance

$$\mathrm{d}(k \mid \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{S}_{\mathrm{pooled}}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_k)' \mathbf{S}_{\mathrm{pooled}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}_k)\right)$$

Mahalanobis Distance

$$\mathrm{d}_{\mathrm{MD}}(k \mid \mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_k)' \mathbf{S}_{\mathrm{pooled}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}_k)$$

Still not optimal:
quadratic in $\mathbf{x}$

$$\mathrm{d}_{\mathrm{MD}} \approx \sum_i \sum_j s_{ij} x_i x_j$$

Combining traces for $n_{\mathrm{a}} > 1$

$$\mathrm{d}_{\mathrm{MD}}^{\mathrm{joint}}(k \mid \mathbf{X}_{k\star}) = -\frac{1}{2} \sum_{\mathbf{x}_i \in \mathbf{X}_{k\star}} (\mathbf{x}_i - \bar{\mathbf{x}}_k)' \mathbf{S}_k^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_k)$$

Combining traces for $n_\mathrm{a} > 1$

$$\mathrm{d}_{\mathrm{MD}}^{\mathrm{joint}}(k \mid \mathbf{X}_{k\star}) = -\frac{1}{2} \sum_{\mathbf{x}_i \in \mathbf{X}_{k\star}} (\mathbf{x}_i - \bar{\mathbf{x}}_k)' \mathbf{S}_k^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_k)$$

- Computation of MD: $O(m^2)$

n_a = number of traces used in attack

Combining traces for $n_{\mathrm{a}} > 1$

$$\mathrm{d}_{\mathrm{MD}}^{\mathrm{joint}}(k \mid \mathbf{X}_{k\star}) = -\frac{1}{2} \sum_{\mathbf{x}_i \in \mathbf{X}_{k\star}} (\mathbf{x}_i - \bar{\mathbf{x}}_k)' \mathbf{S}_k^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_k)$$

- Computation of MD: $O(m^2)$
- Total computation: $O(n_{\mathrm{a}} m^2)$
  - Not good for large m
  - 3 days for $m = 125, n_{\mathrm{a}} = 1000$

n_{a} = number of traces used in attack

Linear Discriminant

$$\mathrm{d}^{\mathrm{joint}}_{\mathrm{LINEAR}}(k \mid \mathbf{X}_{k\star}) = \bar{\mathbf{x}}'_k \mathbf{S}^{-1}_{\mathrm{pooled}} \boxed{\left(\sum_{\mathbf{x}_i \in \mathbf{X}_{k\star}} \mathbf{x}_i \right)} - \frac{n_{\mathrm{a}}}{2} \bar{\mathbf{x}}'_k \mathbf{S}^{-1}_{\mathrm{pooled}} \bar{\mathbf{x}}_k$$

Computation in $O(n_{\mathrm{a}} + m^2)$

- Much better than $\mathrm{d}^{\mathrm{joint}}_{\mathrm{MD}}$: $O(n_{\mathrm{a}} m^2)$
- In practice: for $m = 125, n_{\mathrm{a}} = 1000$
  - $\mathrm{d}^{\mathrm{joint}}_{\mathrm{MD}}$ needs 3 *days*
  - $\mathrm{d}^{\mathrm{joint}}_{\mathrm{LINEAR}}$ only 30 *minutes*

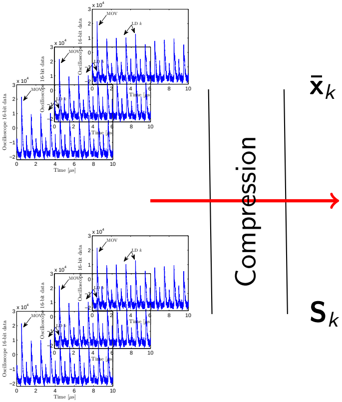
Compression Methods

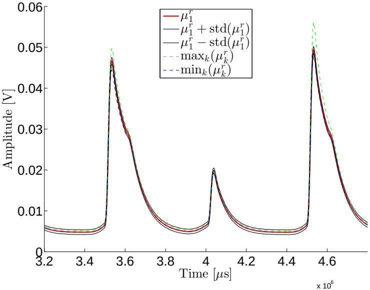
$k = 0$

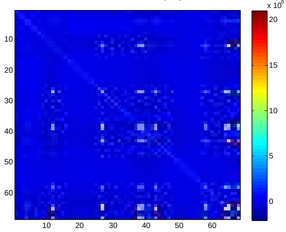
$k = 1$

$\vdots$

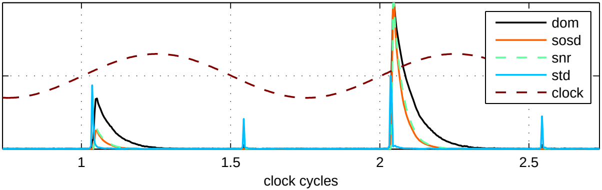
$k = 255$







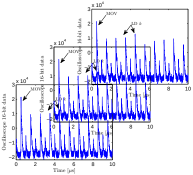
Compression Methods: Sample Selection



Myth: *"Additional samples per clock do not provide additional information"* [Rechberger,Oswald '05]

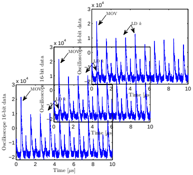
- 1ppc: 1 point per clock [Rechberger,Oswald '05]
- 3ppc (20 samples)
- 20ppc (70 samples)
- allap (125 samples)

Compression Methods: PCA



$$\begin{bmatrix} \mathbf{X}_0^r \\ \mathbf{X}_1^r \\ \vdots \\ \mathbf{X}_{255}^r \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} \rightarrow \mathrm{PCA} \rightarrow \mathbf{U}$$

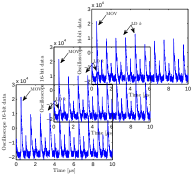
Compression Methods: PCA



$$\begin{bmatrix} \mathbf{X}_0^r \\ \mathbf{X}_1^r \\ \vdots \\ \mathbf{X}_{255}^r \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} \rightarrow \mathrm{PCA} \rightarrow \mathbf{U}$$

[Archambeau et al. '06] $\mathbf{U}' \underset{(\text{large } m)}{\mathbf{S}_k^r} \mathbf{U} = \underset{(\text{small } m)}{\mathbf{S}_k}$

Compression Methods: PCA



$$\begin{bmatrix} \mathbf{X}_0^{\mathrm{r}} \\ \mathbf{X}_1^{\mathrm{r}} \\ \vdots \\ \mathbf{X}_{255}^{\mathrm{r}} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} \rightarrow \mathrm{PCA} \rightarrow \mathbf{U}$$

[Archambeau et al. '06] $\mathbf{U}' \underset{(\text{large } m)}{\mathbf{S}_k^{\mathrm{r}}} \mathbf{U} = \underset{(\text{small } m)}{\mathbf{S}_k}$

Our approach 1. $\underset{(\text{large } m)}{\mathbf{X}_k^{\mathrm{r}}} \mathbf{U} = \underset{(\text{small } m)}{\mathbf{X}_k}$

2. $\mathbf{S}_k = \mathrm{Cov}(\mathbf{X}_k)$

Compression Methods: LDA

$$\begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} + \mathbf{S}_{\text{pooled}} \rightarrow \text{LDA} \rightarrow \mathbf{U}$$

Compression Methods: LDA

$$\begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} + \mathbf{S}_{\text{pooled}} \rightarrow \text{LDA} \rightarrow \mathbf{U}$$

[Standaert et al. '08] $\mathbf{U}' \underset{(\text{large } m)}{\mathbf{S}_k^{\mathrm{r}}} \mathbf{U} = \underset{(\text{small } m)}{\mathbf{S}_k}$

Compression Methods: LDA

$$\begin{bmatrix} \bar{\mathbf{x}}_0 \\ \bar{\mathbf{x}}_1 \\ \vdots \\ \bar{\mathbf{x}}_{255} \end{bmatrix} + \mathbf{S}_{\text{pooled}} \rightarrow \text{LDA} \rightarrow \mathbf{U}$$

[Standaert et al. '08] $\mathbf{U}' \underset{\text{(large } m)}{\mathbf{S}_k^{r}} \mathbf{U} = \underset{\text{(small } m)}{\mathbf{S}_k}$

Our approach: $\mathbf{S}_k = \mathbf{I}$ (we can ignore it, while using all information!)

Evaluation by *Guessing Entropy*

1. Sort candidates by decreasing score $\mathrm{d}(k \mid \mathbf{X}_{k\star})$

$$\begin{array}{ccccc} & & 1 & & k = 74 \\ & & 2 & & k = 13 \\ D_{k\star} & = & 3 & & k = k\star = 9 \\ \text{depth of correct } k & & \vdots & & \vdots \\ & & 256 & & k = 201 \end{array}$$

Evaluation by *Guessing Entropy*

1. Sort candidates by decreasing score $\mathrm{d}(k \mid \mathbf{X}_{k\star})$

$$\begin{array}{ccc} & & 1 \quad\quad k = 74 \\ & & 2 \quad\quad k = 13 \\ D_{k\star} & = & 3 \quad k = k\star = 9 \\ \text{depth of correct } k & & \vdots \quad\quad\quad \vdots \\ & & 256 \quad k = 201 \end{array}$$

2. Compute average over all $k\star$: $\bar{D}_{k\star}$

Evaluation by *Guessing Entropy*

1. Sort candidates by decreasing score $\mathrm{d}(k \mid \mathbf{X}_{k\star})$

$$\begin{array}{ccc} & & 1 & k = 74 \\ & & 2 & k = 13 \\ D_{k\star} & = & 3 & k = k\star = 9 \\ \text{depth of correct } k & & \vdots & \vdots \\ & & 256 & k = 201 \end{array}$$

2. Compute average over all $k\star$: $\bar{D}_{k\star}$
3. *Guessing Entropy* $= \log_2 \bar{D}_{k\star}$

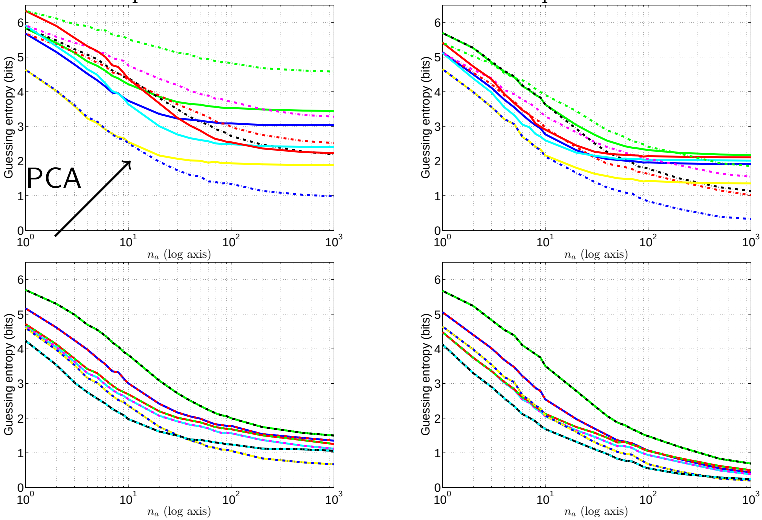
Estimates the remaining *key strength* in targeted brute force search that tries most likely candidates first

Results

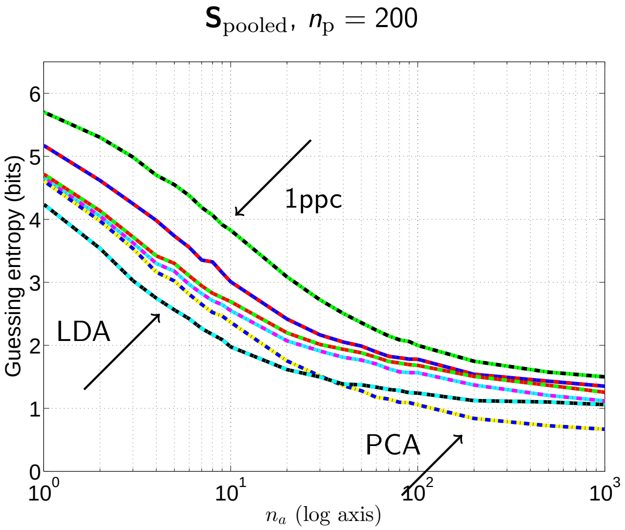
$n_{\mathrm{p}} = 200$ $\qquad$ $n_{\mathrm{p}} = 2000$

$\mathbf{S}_k$

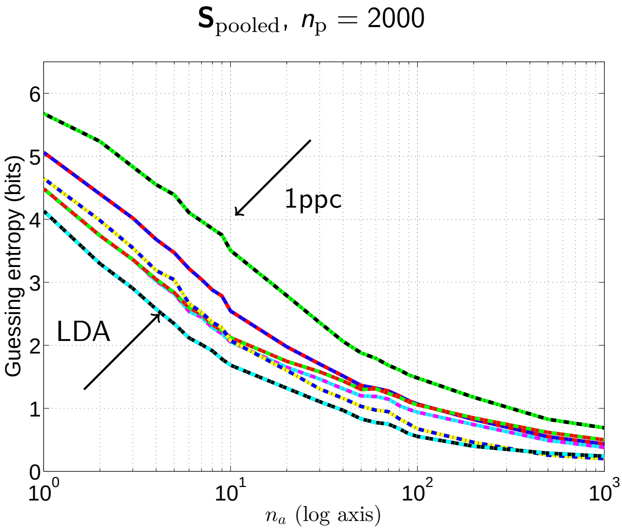
$\mathbf{S}_{\mathrm{pooled}}$



Results

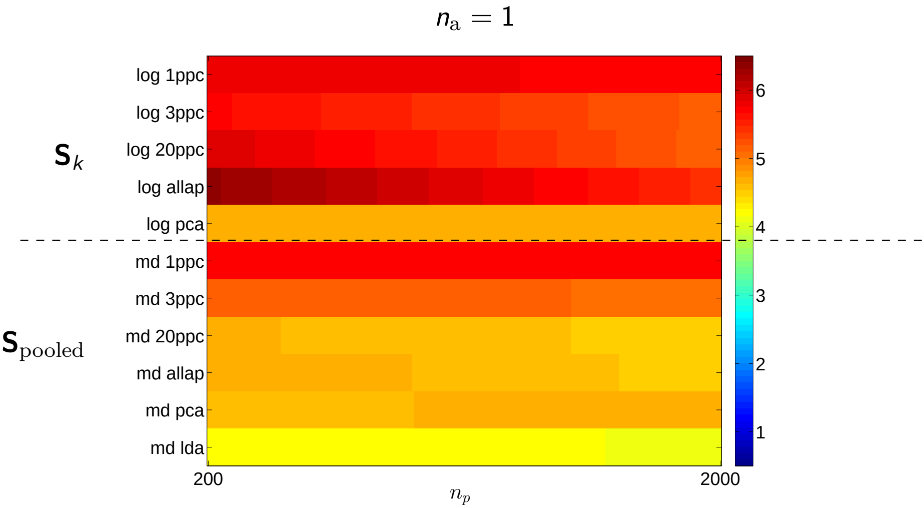

$\mathbf{S}_{\mathrm{pooled}}$, $n_{\mathrm{p}} = 200$
Guessing entropy (bits)
0
1
2
3
4
5
6
10^0
10^1
10^2
10^3
n_a (log axis)
1ppc
LDA
PCA

Results

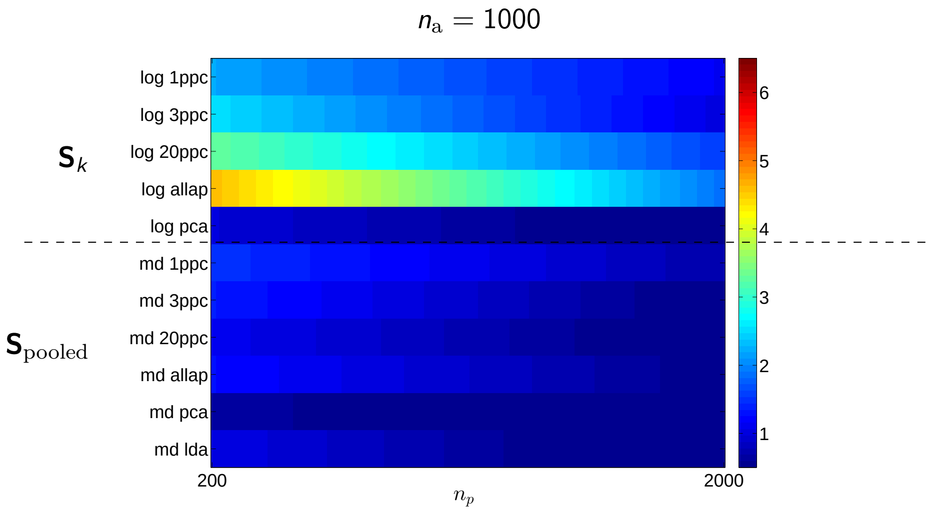


$\mathbf{S}_{\text{pooled}}$, $n_{\text{p}} = 2000$
Guessing entropy (bits)
0
1
2
3
4
5
6
1ppc
LDA
10^0
10^1
10^2
10^3
n_a (log axis)

Practical Guideline



Practical Guideline



Code and Data available

`http://www.cl.cam.ac.uk/research/security/datasets/grizzly/`

- Raw data used for all the results shown in the paper.
- MATLAB scripts to compute template attacks efficiently, including all the algorithms described in the paper.

Conclusion

- Template Attacks can be much more efficient than we thought
  - Can use large number of samples
  - Evaluation time reduced from 3 days to 30 minutes
  - Explore this when using template attacks
  - Might influence CC Evaluation
- Be aware of incorrect assumptions/implementations
  $\Rightarrow$ Now you have our paper!
- Practical guideline for choosing the right compression method
- Now you have data and code to implement efficient template attacks

Questions?

Omar Choudary: omar.choudary@cl.cam.ac.uk

Markus G. Kuhn: markus.kuhn@cl.cam.ac.uk