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+
```

![Amplitude vs Time Graph]

Omar Choudary, Markus G. Kuhn
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

**Executed Code:**
- `movw r30, r24`
- `ld r8, 0`
- `ld r9, k`
- `ld r10, 0`
- `ld r11, 0`

$k = 0$

$k = 1$

$k = 255$
Profiling: Estimate Templates

\(k = 0\)

\(k = 1\)

\(\vdots\)

\(k = 255\)
Attack: using the multivariate normal distribution

\[ d(k \mid x) = \frac{1}{\sqrt{(2\pi)^m |S_k|}} \exp \left( -\frac{1}{2} (x - \bar{x}_k)' S_k^{-1} (x - \bar{x}_k) \right) \]

\[ k^* \rightarrow \text{argmax}_k d(k \mid x) \]
Problem 1: Floating point issues

\[ d(k \mid x) = \frac{1}{\sqrt{(2\pi)^m |S_k|}} \exp \left( -\frac{1}{2} (x - \bar{x}_k)'S_k^{-1}(x - \bar{x}_k) \right) \]

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

\[ d(k \mid x) = \frac{1}{\sqrt{(2\pi)^m |S_k|}} \exp \left( -\frac{1}{2} (x - \bar{x}_k)' S_k^{-1} (x - \bar{x}_k) \right) \]

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

These are real problems. Naive implementations are likely to fail.
Solution: use LOG

\[ d_{\text{LOG}}(k \mid x) = -\frac{m}{2} \log 2\pi - \frac{1}{2} \log |S_k| - \frac{1}{2} (x - \bar{x}_k)' S_k^{-1} (x - \bar{x}_k) \]
Caveat: pdf can be larger than 1

"[Choose the candidate \( k \) that leads to the] smallest \( \text{absolute value} \) [of \( d_{\text{LOG}} \)]"

[Mangard, Oswald, Popp '07]
Caveat: pdf can be larger than 1

"[Choose the candidate $k$ that leads to the] smallest absolute value [of $d_{\text{LOG}}$]"

Incorrect:

log is monotonic, abs is not!
We choose $k$ with highest value of $d_{\text{LOG}}$.

[Mangard, Oswald, Popp '07]
Problem 2: dealing with large number of samples

- Myth: problems with inversion of $S_k$ as soon as $m$ is large.

$$m = \text{number of samples}$$

$$n_p = \text{number of traces from profiling, for each } k$$
Problem 2: dealing with large number of samples

- Myth: problems with inversion of $S_k$ as soon as $m$ is large.
- Clarification:
  - $n_p \leq m$: $S_k$ cannot be inverted ($\text{rank}(S_k) < n_p$)

$m =$ 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 $S_k$ as soon as $m$ is large.
- Clarification:
  - $n_p \leq m$: $S_k$ cannot be inverted ($\text{rank}(S_k) < n_p$)
  - $n_p > m$: $S_k$ 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 $S_k$ as soon as $m$ is large.
- Clarification:
  - $n_p \leq m$: $S_k$ cannot be inverted ($\text{rank}(S_k) < n_p$)
  - $n_p > m$: $S_k$ will most likely be invertible (ignoring highly correlated samples)
- Problem: obtaining $n_p > m$ can be difficult due to memory and time constraints.

$m = \text{number of samples}$

$n_p = \text{number of traces from profiling, for each } k$
Scenario 1: $S_k$ dependent on $k$
Scenario 2: $S_k$ independent on $k$
Efficient solution: use $S_{\text{pooled}}$

- $S_{\text{pooled}}$ is an average of the covariances.
- $S_{\text{pooled}}$ uses $|S|n_p$ traces, while $S_k$ only $n_p$.
- Now the condition for non-singularity is $n_p > \frac{m}{|S|}$
  - A great advantage in practice.
Mahalanobis Distance

\[ d(k \mid x) = \frac{1}{\sqrt{(2\pi)^m|S_{\text{pooled}}|}} \exp \left( -\frac{1}{2} (x - \bar{x}_k)' S_{\text{pooled}}^{-1} (x - \bar{x}_k) \right) \]
Mahalanobis Distance

\[ d_{MD}(k \mid x) = -\frac{1}{2}(x - \bar{x}_k)'S^{-1}_{\text{pooled}}(x - \bar{x}_k) \]

Still not optimal:

Still quadratic in \( x \)

\[ d_{MD} \approx \sum_i \sum_j s_{ij} x_i x_j \]
Combining traces for $n_a > 1$

$$d_{\text{MD}}^{\text{joint}}(k \mid X_{k*}) = -\frac{1}{2} \sum_{x_i \in X_{k*}} (x_i - \bar{x}_k)'S_k^{-1}(x_i - \bar{x}_k)$$
Combining traces for $n_a > 1$

\[
d_{\text{MD}}^{\text{joint}}(k \mid X_{k^{*}}) = -\frac{1}{2} \sum_{x_i \in X_{k^{*}}} (x_i - \bar{x}_k)' S_k^{-1} (x_i - \bar{x}_k)
\]

- Computation of MD: $O(m^3)$

$n_a = \text{number of traces used in attack}$
Combining traces for $n_a > 1$

\[
d_{\text{MD}}^{\text{joint}}(k \mid X_{k^*}) = -\frac{1}{2} \sum_{x_i \in X_{k^*}} (x_i - \bar{x}_k)' S_k^{-1}(x_i - \bar{x}_k)
\]

- Computation of MD: $O(m^3)$
- Total computation: $O(n_a m^3)$
  - Not good for large $m$
  - 3 days for $m = 125, n_a = 1000$

$n_a = \text{number of traces used in attack}$
Linear Discriminant

\[ d_{\text{LINEAR}}^{\text{joint}}(k \mid X_{k^*}) = \bar{x}_k' S_{\text{pooled}}^{-1} \left( \sum_{x_i \in X_{k^*}} x_i \right) - \frac{n_a}{2} \bar{x}_k' S_{\text{pooled}}^{-1} \bar{x}_k \]

Computation in \( O(n_a + m^3) \)

- Much better than \( d_{\text{MD}}^{\text{joint}} : O(n_a m^3) \)
- In practice: for \( m = 125, n_a = 1000 \)
  - \( d_{\text{MD}}^{\text{joint}} \) needs 3 days
  - \( d_{\text{LINEAR}}^{\text{joint}} \) only 30 minutes
**Compression Methods**

$k = 0$

$k = 1$

$\vdots$

$k = 255$

**Compression**

$\bar{x}_k$

$S_k$
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

\[
\begin{bmatrix}
X^r_0 \\
X^r_1 \\
\vdots \\
X^r_{255}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix}
\rightarrow \text{PCA} \rightarrow U
\]
Compression Methods: PCA

\[
\begin{bmatrix}
X_0^r \\
X_1^r \\
\vdots \\
X_{255}^r
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix}
\rightarrow \text{PCA} \rightarrow U
\]

\[U' \quad S_k^r \quad U = S_k \quad \text{(large } m) \quad \text{(small } m)\]

[Archambeau et al. ’06]
Compression Methods: PCA

\[
\begin{bmatrix}
X^r_0 \\
X^r_1 \\
\vdots \\
X^r_{255}
\end{bmatrix} \rightarrow 
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix} \rightarrow \text{PCA} \rightarrow \mathbf{U}
\]

[Archambeau et al. '06]

\[
\mathbf{U}' S^r_k \mathbf{U} = S^r_k
\]

(large \(m\))

(small \(m\))

Our approach

1. \(X^r_k \mathbf{U} = X^r_k\)

(large \(m\))

(small \(m\))

2. \(S_k = \text{Cov}(X_k)\)
Compression Methods: LDA

\[
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix} + S_{\text{pooled}} \rightarrow \text{LDA} \rightarrow \mathbf{U}
\]
Compression Methods: LDA

\[
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix}
+ S_{\text{pooled}} \rightarrow \text{LDA} \rightarrow U
\]

[Standaert et al. '08] \quad U' \quad S_k^r \quad U = S_k

\text{(large } m) \quad \text{(small } m)
Compression Methods: LDA

\[
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix} + S_{\text{pooled}} \rightarrow \text{LDA} \rightarrow U
\]

[Standaert et al. ’08] \quad U' \quad S^r_k \quad U = S_k \\
(large \ m) \quad (small \ m)

Our approach: \( S_k = I \) (we can ignore it, while using all information!)
1. Sort candidates by decreasing score $d(k \mid X_{k^*})$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$k = 74$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$k = 13$</td>
<td></td>
</tr>
<tr>
<td>$D_{k^*}$</td>
<td>=</td>
<td>3</td>
</tr>
<tr>
<td>$k = k^* = 9$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth of correct $k$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>$k = 201$</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation by *Guessing Entropy*

1. Sort candidates by decreasing score $d(k \mid X_{k*})$

   $\begin{array}{c|c}
   \text{depth of correct } k & \text{value} \\
   \hline
   1 & k = 74 \\
   2 & k = 13 \\
   3 & k = k* = 9 \\
   \vdots & \vdots \\
   256 & k = 201 \\
   \end{array}$

2. Compute average over all $k*$: $\overline{D}_{k*}$
Evaluation by *Guessing Entropy*

1. Sort candidates by decreasing score $d(k \mid X_{k^*})$

   - $k = 74$
   - $k = 13$
   - $k^* = 9$
   - $k = 201$

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

3. *Guessing Entropy* = $\log_2 D_{k^*}$

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

$n_P = 200$

$n_P = 2000$

$S_k$

PCA

$S_{pooled}$

$n_a$ (log axis)

Guessing entropy (bits)
Results

\( S_{\text{pooled}}, n_p = 200 \)

![Graph showing guessing entropy for different methods: LDA, PCA, and 1ppc.](image)

- **LDA**
- **PCA**
- **1ppc**
Results

\[ S_{\text{pooled}}, n_p = 2000 \]
Practical Guideline

$n_a = 1$

$S_k$

$S_{pooled}$

Omar Choudary, Markus G. Kuhn
Efficient Template Attacks
Slide 36
Practical Guideline

$n_a = 1000$

$S_k$

$S_{pooled}$

$n_p$

200 2000

log 1ppc
log 3ppc
log 20ppc
log allap
log pca
md 1ppc
md 3ppc
md 20ppc
md allap
md pca
md lda
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
  ⇒ 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