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)
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:

\[
\text{movw r30, r24} \\
\text{ld r8, Z+} \\
\text{ld r9, Z+} \\
\text{ld r10, Z+} \\
\text{ld r11, Z+}
\]
Experiment: eavesdropping on 8-bit data bus

Executed Code:

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

Omar Choudary, Markus G. Kuhn

Efficient Template Attacks

Slide 4
Profiling: Acquire Traces

**Executed Code:**

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

$k = 0$

$k = 1$

\[ \vdots \]

$k = 255$
Profiling: Estimate Templates

$k = 0$

$k = 1$

$\vdots$

$k = 255$
**Attack: using the multivariate normal distribution**

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

\[k^* \rightarrow \mathop{\text{argmax}}_k d(k \mid x)\]
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 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 ($\text{rank}(S_k) < n_p$)

$m = \text{number of samples}$
$n_p = \text{number of traces from profiling, for each } k$
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 ($\text{rank}(S_k) < n_p$)
  - $n_p > m$: $S_k$ will most likely be invertible
    (ignoring highly correlated samples)

$m = \text{number of samples}$

$n_p = \text{number of traces from profiling, for each } k$
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 ($\text{rank}(S_k) < n_p$)
  - $n_p > m$: $S_k$ will most likely be invertible
    (ignoring highly correlated samples)
- Problem: obtaining $n_p > m$ can be difficult due to memory and time constraints.

$m =$ number of samples

$n_p =$ 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

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

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

Still not optimal:
quadratic in \( x \)

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

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

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

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

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

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

- Computation of MD: $O(m^2)$
- Total computation: $O(n_a m^2)$
  - Not good for large $m$
  - 3 days for $m = 125, n_a = 1000$

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

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

**Computation in** \( O(n_a + m^2) \)

- Much better than \( d_{\text{MD}}^\text{joint} : O(n_a m^2) \)
- 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

\[ k = 0 \]

\[ k = 1 \]

\ldots

\[ k = 255 \]
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

\[
\begin{bmatrix}
X_r^0 \\
X_r^1 \\
\vdots \\
X_r^{255}
\end{bmatrix} \rightarrow
\begin{bmatrix}
\bar{x}_0 \\
\bar{x}_1 \\
\vdots \\
\bar{x}_{255}
\end{bmatrix} \rightarrow \text{PCA} \rightarrow U
\]

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

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

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

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

Our approach

1. \[\mathbf{X}_k^r \quad \mathbf{U} = \mathbf{X}_k \quad \text{(large } m \text{)} \quad \text{(small } m \text{)}\]
2. \[\mathbf{S}_k = \text{Cov}(\mathbf{X}_k)\]

[Archambeau et al. '06]
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

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

\[ U' \quad S_k^r \quad U = S_k \]

[Standaert et al. ’08]

\( U' \) (large \( m \)) \quad \( S_k^r \) \quad \( U \) = \( S_k \) (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] \( U' S_k^r U = S_k \)

(\text{large } m) \hspace{1cm} (\text{small } m)

Our approach: \( S_k = I \) (we can ignore it, while using all information!)
Evaluation by *Guessing Entropy*

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

   $\begin{align*}
   &1 \quad k = 74 \\
   &2 \quad k = 13 \\
   &D_{k^*} = 3 \quad k = k^* = 9 \\
   &\vdots \quad \vdots \\
   &256 \quad k = 201
   \end{align*}$
Evaluation by *Guessing Entropy*

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

\[
\begin{align*}
1 & \quad k = 74 \\
2 & \quad k = 13 \\
D_{k^*} & = 3 \quad k = k^* = 9 \\
depth of correct k & : : \\
256 & \quad k = 201
\end{align*}
\]

2. Compute average over all \( k^* \): \( \bar{D}_{k^*} \)
Evaluation by *Guessing Entropy*

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

   \[
   \begin{align*}
   &1 & k = 74 \\
   &2 & k = 13 \\
   \end{align*}
   \]

   \[D_{k^*} = 3 \quad k = k^* = 9\]

   depth of correct $k$

   \[
   \begin{align*}
   &\vdots \quad \vdots \\
   &256 & k = 201
   \end{align*}
   \]

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

3. *Guessing Entropy* $= \log_2 \overline{D}_{k^*}$

   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
Results

\[ S_{\text{pooled}}, \ n_p = 200 \]

Guessing entropy (bits)

\( n_a \) (log axis)

LDA

PCA

1ppc
$S_{\text{pooled}}, n_p = 2000$

Guessing entropy (bits)

$n_a$ (log axis)

LDA

1ppc
Practical Guideline

$n_a = 1$

$S_k$

$S_{pooled}$

$n_p$

200

2000

$\log 1ppc$

$\log 3ppc$

$\log 20ppc$

$\log \text{allap}$

$\log \text{pca}$

$\text{md } 1ppc$

$\text{md } 3ppc$

$\text{md } 20ppc$

$\text{md } \text{allap}$

$\text{md } \text{pca}$

$\text{md } \text{lda}$
Practical Guideline

$n_a = 1000$

$S_k$

$S_{pooled}$
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