Leakage Detection with the $\chi^2$-Test

Amir Moradi$^1$, Bastian Richter$^1$, Tobias Schneider$^2$ and François-Xavier Standaert$^2$

$^1$ Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
$^2$ ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium

CHES 2018, Amsterdam 10.09.2018
• How to assure that a device does not leak sensitive values during execution of a cryptographic operation?

• Often performed based on attacks (e.g. in Common Criteria)
  – High complexity
  – Every attack has to be optimized
  – Easy to miss an attack vector
Motivation
Leakage Detection

• General approach to detect leakage independent of models or attack methods

• Reduction to general statistical assumptions without a specific model for the implementation (black box)

TVLA based on Welch‘s $t$-test [1]:

1. Reduction to two classes (e.g. fixed-vs.-random)
2. Simple statistical treatment (estimation of statistical moments)

Motivation

t-Test Problems

• The two properties can lead to problems

1. Reduction to two classes
   – False negative because of leakage which is too similar in two classes but would be detectable with more classes
Motivation

$t$-Test Problems

• The two properties can lead to problems

1. Reduction to two classes
   – False negative because of leakage which is too similar in two classes but would be detectable with more classes

2. Estimation and comparison of separate moments
   – False negative because of leakage distributed over multiple moments
Motivation

$t$-Test Problems

- The two properties can lead to problems

1. Reduction to two classes
   - **False negative** because of leakage which is too similar in two classes but would be detectable with more classes

\[ \chi^2 \text{-Test works with multiple classes} \]

2. Estimation and comparison of separate moments
   - **False negative** because of leakage distributed over multiple moments
Motivation

$t$-Test Problems

- The two properties can lead to problems

1. Reduction to two classes
   - False negative because of leakage which is too similar in two classes but would be detectable with more classes

\[ \chi^2 \text{-Test works with multiple classes} \]

2. Estimation and comparison of separate moments
   - False negative because of leakage distributed over multiple moments

\[ \chi^2 \text{-Test is based on the whole distribution} \]
\( \chi^2 \)-Test Methodology

Fixed vs. Random

1. Measure traces for random or fixed input in random order
\( \chi^2 \)-Test Methodology

Fixed vs. Random

1. Measure traces for random or fixed input in random order

2. Compute histograms for each point of classes
**χ²-Test Methodology**

Fixed vs. Random

1. Measure traces for random or fixed input in random order

2. Compute histograms for each point of classes

3. Compute contingency table $F_{i,j}$ from histograms
\( \chi^2 \)-Test Methodology

**Fixed vs. Random**

1. Measure traces for random or fixed input in random order

2. Compute histograms for each point of classes

3. Compute contingency table \( F_{i,j} \) from histograms

4. Compute \( x, v, \) and \( p \) from table \( F_{i,j} \)
**$\chi^2$-Test Methodology**

**Fixed vs. Random**

1. Measure traces for random or fixed input in random order

2. Compute histograms for each point of classes

3. Compute contingency table $F_{i,j}$ from histograms

4. Compute $x$, $v$, and $p$ from table $F_{i,j}$

---

**Same procedure as for $t$-test** [1]

---

[1] Reparaz et al., Fast Leakage Assessment, CHES 2017
\( \chi^2 \)-Test Methodology

Fixed vs. Fixed

1. Measure traces for \( r \) different inputs in random order

2. Compute histograms for each point of classes

3. Compute contingency table \( F_{i,j} \) from histograms

4. Compute \( x \), \( v \), and \( p \) from table \( F_{i,j} \)

\[
x = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(F_{i,j} - E_{i,j})^2}{E_{i,j}}
\]

\[
p = \int_x^\infty f(x, v) \, dx,
\]
**χ²-Test**

Pearson's $\chi^2$-Test of Independence

- **Null hypothesis:** The occurrences of the observations are independent.

- If the test concludes that the null hypothesis is rejected, the leakage is assumed to be informative.

- Evaluation of independence based on contingency table of frequencies.

- In contrast to the $t$-test we have to calculate the p-values for the $\chi^2$-test as the degree-of-freedom does not converge.
  - We chose $p = 10^{-5}$ as threshold (equivalent to $t = 4.5$).
1. Build contingency table $F_{i,j}$ from histograms
**χ²-Test**

Pearson’s $\chi^2$-Test of Independence

1. Build contingency table $F_{i,j}$ from histograms

2. Calculate expected values $E_{i,j}$ for each cell

\[
E_{i,j} = \frac{\left(\sum_{k=0}^{c-1} F_{i,k}\right) \cdot \left(\sum_{k=0}^{r-1} F_{k,j}\right)}{N}
\]

\[
E_{0,0} = \frac{(24 + 59 + 28 + 9) \cdot (24 + 25)}{220} = \frac{120 \cdot 47}{220} \approx 25.64
\]

<table>
<thead>
<tr>
<th>$F_{i,j}$</th>
<th>$j = 0$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 0$</td>
<td>24</td>
<td>59</td>
<td>28</td>
<td>9</td>
<td>120</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>23</td>
<td>57</td>
<td>20</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>total</td>
<td>47</td>
<td>116</td>
<td>48</td>
<td>9</td>
<td>220</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$E_{i,j}$</th>
<th>$j = 0$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 0$</td>
<td>25.64</td>
<td>63.18</td>
<td>26.18</td>
<td>4.91</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>21.36</td>
<td>52.73</td>
<td>21.82</td>
<td>4.09</td>
</tr>
</tbody>
</table>
3. Calculate $\chi^2$-test statistic $x$ and degree-of-freedom $v$

$$x = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(F_{i,j} - E_{i,j})^2}{E_{i,j}}$$

$$v = (2 - 1) \cdot (4 - 1) = 3$$

For cell (0,0):

$$\frac{(24 - 25.64)^2}{25.64} \approx 0.10$$

$$x = 0.10 + 0.29 + 0.13 + 3.41 + 0.13 + 0.35 + 0.15 + 4.09 = 8.64$$
3. Calculate $\chi^2$-test statistic $x$ and degree-of-freedom $\nu$

$$x = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(F_{i,j} - E_{i,j})^2}{E_{i,j}}$$

$$\nu = (2 - 1) \cdot (4 - 1) = 3$$

For cell (0,0): $\frac{(24 - 25.64)^2}{25.64} \approx 0.10$

$$x = 0.10 + 0.29 + 0.13 + 3.41 + 0.13 + 0.35 + 0.15 + 4.09 = 8.64$$

4. Derive $p$ value using the $\chi^2$ probability density function

$$p = \int_{x}^{\infty} f(x, \nu) \, dx$$

$$f(x, \nu) = \begin{cases} \frac{\nu^{\frac{\nu}{2} - 1} e^{-\frac{x}{2}}}{\frac{\nu}{2}^{\nu} 2^{\nu} \Gamma\left(\frac{\nu}{2}\right)}, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$p \approx 0.0345$$
Simulated Experiments
Univariate

- Simulation of masked hardware design with parallel processing of $d$ shares
- Secret value $X$ is split into $d$ Boolean shares $X_i$

$$X = X_0 \oplus X_1 \oplus \ldots \oplus X_{d-1}$$

- Leakage is combined with Hamming Weight leakage function with additive Gaussian noise for three different SNRs

$$L = \sum_{i=0}^{d-1} HW(X_i) + \mathcal{N}_{0,\sigma}$$

- Traces are generated for Fixed vs. Random test
Simulated Experiments
Univariate Results - Orders

- $t$-test significantly outperforms $\chi^2$-test for lower orders ($d = 1, 2$)

- $\chi^2$-test improves with higher orders and is significantly better in order $d = 4$

- Advantage is expected to increase with higher orders
Simulated Experiments
Univariate Results - SNR

- Results of $\chi^2$-test are highly affected by the SNR
- For low SNRs the advantage in higher orders disappears
- A high SNR significantly improves the detection of leakage
- Effect of SNR on $t$-test is much lower

\[ d = 3, \text{SNR}_1 = 0.1 \]
\[ d = 3, \text{SNR}_2 = 1.0 \]
\[ d = 3, \text{SNR}_3 = 10 \]
Simulated Experiments
Multivariate

- Simulation of software or serialized hardware masking

- Leakage of different shares at separate points in time

\[ L_{t_i} = HW(X_i) + N_{0,\sigma}, \quad 0 \leq i < d \]

- We evaluated three different options to combine leakage

- Normalized Product: (\(\chi^2\)- and t-test)

\[
L' = \prod_{i=0}^{d-1} (L_{t_i} - \mu_{t_i})
\]
Simulated Experiments
Multivariate

• **Sum Combining:**
  Possible because whole distribution and not only the means are compared

\[
L' = \sum_{i=0}^{d-1} L_{ti}
\]

+ Noise Terms are not multiplied

• **Multivariate Histograms:**

\[
L' = (L_{t_0}, L_{t_1}, ..., L_{t_{d-1}})
\]
Simulated Experiments
Multivariate Results

- Unless for very high SNRs, $t$-test works better than the $\chi^2$-test
- The normalized product works best for all orders with non-negligible noise
- Sum combining and multivariate histograms only improve the results for very low noise

$d = 2, \text{SNR}_2 = 1.0$

$d = 3, \text{SNR}_2 = 1.0$

$d = 4, \text{SNR}_2 = 1.0$

$d = 4, \text{SNR}_4 = 20.0$
Experiments

Target

• Threshold Implementation of PRESENT with 3 shares
• S-box split up into two functions $G$ and $F$
• Byte-serial implementation with shift register for state

• Implemented on Spartan-6 (SAKURA-G)
• Running at 160 MHz and measured at 1 GS/s
Experiments
Fixed vs. Random

- As expected leakage detected in orders $d \geq 2$ with main leakage in third order
- $\chi^2$-test shows similar shape as 3rd-order $t$-test
- Confidence is significantly higher for $\chi^2$-test
Experiments
Fixed vs. Fixed

- Traces recorded for eight different fixed plaintexts
- $\chi^2$-test can process the plaintexts as eight classes
- Main leakage at beginning similar to fixed vs. random
- Detects leakage at late times with lower confidence as for the pairs of plaintexts
Experiments
Distinguisher

• Use multi-class capability as distinguisher with model

• For each key candidate $k$
  1. Sort traces into classes by model (e.g. HD)
  2. Calculate histograms for the classes
  3. Calculate $x$, $v$ and $p$

• Rank key candidates by $p$-value
Experiments
Distinguisher

- Utilizes leakage in the whole distribution and not only a single moment

- Similar to Mutual Information Analysis (MIA) but provides a confidence level for each key candidate

- Number of classes has to be lower than number of key candidates (same for MIA)
Experiments

Distinguisher

- CPA and $\chi^2$-test with HD-Model of consecutive S-boxes
- None of the higher-order CPAs is successful (with 50M traces)
- $\chi^2$-test is successful after 28 million traces
• We presented the $\chi^2$-test as a complement to the $t$-test

• It is able to outperform the $t$-test if:
  – The noise level is not sufficient
  – The leakage is distributed over multiple statistical moments

• It should only be used together with the $t$-test, since there are cases which are not detected.

• Use $t$-test to evaluate the security order and $\chi^2$-test to evaluate the noise level.
Thank You For Your Attention!
Any Questions?
Experiments
Fixed vs. Fixed

- Traces recorded for eight different fixed plaintexts
- Five combinations of two plaintexts plotted
- Different combinations detect leakage at different times
- Third order $t$-test and $\chi^2$-test again similar
- $\chi^2$-test again gives higher confidence
Experiments
Performance

• $t$-Test and $\chi^2$-Test implemented in C++

• Based on histograms computed before for both tests

• Both tests need approx. 2.8 µs per point on an Intel i7-6600U @2.6GHz

• Calculation of $t$-Test only speeds up by 0.4 µs when omitting the calculation of p value and degree of freedom