Exploiting Small Leakages in Masks to Turn a Second-Order Attack into a First-Order Attack

Alexander DeTrano
Sylvain Guilley
Xiaofei Guo
Naghmeh Karimi
Ramesh Karri

HASP 2015
Motivation
Side-Channel Attacks

- Algorithm implementations may inadvertently leak information through different sources
- These sources are called “side-channels”
- A side-channel attack exploits one or more of these to learn secret information
Masking Countermeasure

• Countermeasures such as masking have been developed to thwart side-channel attacks
• Masking tries to remove the correlation between the power consumption and the data that is being handled

Boolean Masking

<table>
<thead>
<tr>
<th>Unmasked</th>
<th>Masked</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = d \oplus k$</td>
<td>$x \downarrow m = d \oplus k \oplus m$</td>
</tr>
</tbody>
</table>

d : plaintext, k : secret key, m : uniformly distributed random mask
Advanced Encryption Standard (AES)

• Plaintext/ciphertext: 128 bits
• Key: 128/192/256 bits
• 13 rounds + 1 pre-round

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoundKey</td>
<td>XOR operation</td>
</tr>
<tr>
<td>SubBytes</td>
<td>Look-up table</td>
</tr>
<tr>
<td>ShiftRows</td>
<td>Byte-wise permutation</td>
</tr>
<tr>
<td>MixColumns</td>
<td>Matrix multiplication</td>
</tr>
</tbody>
</table>
Rotating S-Box Masking (RSM)

- Carefully chosen masks reduces storage requirements
  - RSM uses 16 masks
- HW : 2\textsuperscript{nd}-order zero-offset resistance
- EM traces publicly from DPA Contest v4
  - AES256-RSM implemented on a smartcard with 8-bit microcontroller Atmel ATMega-163

Maxime Nassar, Youssef Souissi, Sylvain Guilley, Jean-Luc Danger. RSM: a Small and Fast Countermeasure for AES, Secure against 1st and 2nd-order Zero-Offset SCAs, DATE’12
Mask Recovery Attack

• A 1st order CPA attack fails to recover the key after 100,000 traces
• Prior work: non-uniform distribution of the masks after an XOR (174), collision attacks (1100), 2nd-order CPA (300),
• Our attack : 10 traces

<table>
<thead>
<tr>
<th>Observation</th>
<th>Idea</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Masks are deployed in a predictable sequence</td>
<td>• Launch a 1st order horizontal CPA attack to recover the masks</td>
</tr>
<tr>
<td>• The device leaks the Hamming Weight of the masks each time they are handled</td>
<td>• Recover the masks, then recover the key</td>
</tr>
</tbody>
</table>

1\textsuperscript{st} order CPA Attack

Hypothetical Power Model

\[ HW(\ldots m0 \; m1 : \ldots m1m2 : \ldots m2m3 : \ldots m4 \ldots m15) \]

Measured Power Traces

[Graph showing power traces over time]

Correlation
Mask Recovery – When does masking take place?

$NICV = \frac{\text{Var}(\mathbb{E}[T|X])}{\text{Var}(T)}$

$W$ : window when masking is suspected to occur

$NICV$ : Normalized Inter-class Variance

$T$ : power traces

$X$ : plaintext byte
Mask Recovery – When does masking take place?

Hypothetical Power Model

$HW(Mask\ Matrix)$

Correlation

Measured Power Traces

$w$: window when masking is guessed to start

$\Delta$: time between similar operations
Mask Recovery Results

Mask Recovery Success Rate

Taken over 10,000 power traces
2\textsuperscript{nd}-order CPA Attack

Hypothetical Power Model

\[ HW(m_0 m_1 m_2 m_3 \ldots m_{15}) \]

Measured Power Traces

Apply combination function

Pre-processed Traces

1\textsuperscript{st} order CPA

Correlation
Comparison with 2\textsuperscript{nd}-Order Attack\(^^\)

Key Bytes Recovered : SNR = 2.689

\begin{align*}
    SNR = \frac{1}{\text{NICV}} - 1 &= \frac{\text{Var}(\mathbb{E}[T|X])}{\text{Var}(T)} - \frac{\text{Var}(\mathbb{E}[T|X])}{\text{Var}(\mathbb{E}[T|X])}
\end{align*}

\( T \) : power traces, \( X \) : plaintext byte

\(^*\)S. Bhasin, J-L Danger, S. Guilley, and Z. Najm, “Side-Channel Leakage and Trace Compression using Normalized Inter-Class Variance”, HASP'14

\(^^\)E. Prouff, M. Rivain, and R. Bevan. Statistical analysis of second order differential power analysis. IEEE Trans. on Computers'09
Adding Noise to the Power Traces

Key Bytes Recovered

Number of Traces

$*SNR = 1 \times NICV - 1 = \frac{\text{Var}(E[T|X])}{\text{Var}(T)} - \frac{\text{Var}(E[T|X])}{\text{Var}(E[T|X])}$

T : power traces, X : plaintext byte

*S. Bhasin, J-L Danger, S. Guilley, and Z. Najm, “Side-Channel Leakage and Trace Compression using Normalized Inter-Class Variance”, HASP’14
Conclusion

• Our attack outperforms a 2\textsuperscript{nd}-order attack by two orders of magnitude w.r.t to number of traces needed to recover the key.

• A 2\textsuperscript{nd}-order attack fails to recover the key for SNR < 0.289, while our attack succeeds for SNR ≤ 0.035.

• The implementation leaks the Hamming Weight of the masks as they are fetched from memory.

• The predictable deployment order of the masks and Hamming Weight variation allow an attacker to recover the mask offset.

• We also analyzed the relationship between mask recovery success rate and window width/number of masks attacked.
Thank you!

Questions?