Analysis and Improvement of the Random Delay Countermeasure of CHES 2009

Jean-Sébastien Coron    Ilya Kizhvatov

CHES 2010, Santa-Barbara, CA, USA
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Random Delays: In Brief

- Algorithm execution
- Target operation

Time
Random Delays: In Brief

- Algorithm execution
- Target operation
- Delay

Time
Random Delays: In Brief

- Algorithm execution
- Target operation
- Delay

Time
Random Delays: In Brief

- Algorithm execution
- Target operation
- Delay

Time
Random Delays: In Brief

Effect in DPA
Random Delays: More Details

\[ S_N = \sum_{i=0}^{N} d_i \]

Assumptions

- multiple delays are harder to remove than a single one
- adversary is facing the cumulative sum of \( N \) delays

Desired properties of \( S_N \)

- should increase attacker’s uncertainty
- smaller mean to decrease performance penalty
Methods with Independent Delay Generation

- Plain uniform delays: $d_i \sim \mathcal{U}[0, a]$
- WISTP07: uniform $\rightarrow$ pit-shaped to increase $\sigma$

Central Limit Theorem: $S_N \xrightarrow{N} \mathcal{N}(N\mu, N\sigma^2)$
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Method of CHES’09: Floating Mean

Idea: generate delays non-independently

Algorithm

- within an execution: generate delays within a small interval $[m, m + b]$
- across executions: vary $m$ within a larger interval $[0, a - b]$
- parameters $a$ and $b$ are fixed for an implementation
Method of CHES’09: Floating Mean

\[ E(S_N) = \frac{Na}{2}, \quad \text{Var}(S_N) = N^2 \cdot \frac{(a - b + 1)^2 - 1}{12} + N \cdot \frac{b^2 + 2b}{12} \]

- **a** = 255 \quad \text{PU}
- **b** = 50 \quad \text{FM}
Method of CHES’09: Floating Mean

\[ E(S_N) = \frac{Na}{2}, \quad \text{Var}(S_N) = N^2 \cdot \frac{(a - b + 1)^2 - 1}{12} + N \cdot \frac{b^2 + 2b}{12} \]

\[ a = 255 \quad \text{PU} \]
\[ b = 50 \quad \text{FM} \]
The Issue with Floating Mean

Using parameters from the practical implementation of CHES'09:

\[ a = 18 \]
\[ b = 3 \]

- cogs are not good for security
- \( \sigma \) is not a good measure of security
The Issue with Floating Mean

Explanation

- $S_N$ is a mixture of $a - b + 1$ Gaussians with means $N \cdot (m + b/2)$ and variance $\sigma^2 \approx Nb^2$
- The distance between component means is $N$
- Components are not visible if $\sigma > N$, which yields the condition
  $$b \gg \sqrt{N}$$

Conclusion

- we have to use longer and less frequent delays in Floating Mean
- this is not good for security and performance
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Improved Floating Mean

Algorithm

1. in an implementation, fix parameters $a$, $b$, and an additional parameter $k$
2. before an execution, generate random $m'$ from $[0, (a - b) \cdot 2^k[$
3. throughout the execution, generate delays $d$ in two steps:
   - generate $d' \in [m', m' + (b + 1) \cdot 2^k[$
   - let $d \leftarrow \lfloor d' \cdot 2^{-k} \rfloor$.

Can be efficiently implemented in 8-bit assembly.
Improved Floating Mean: Distribution

\[
E[S_N] = N \cdot \left( \frac{a}{2} - 2^{-k-1} \right), \quad \text{Var}(S_N) \approx N^2 \cdot \frac{(a - b)^2 - 1}{12}
\]

\(a = 18\) \(\quad k = 3\)

\(b = 3\)

![Graph showing comparison between IFM and FM with 32 delays]
Condition on Parameters

Cogs are not visible when

\[ b \gg \sqrt{N} \cdot 2^{-k} \]

⇒ shorter and more frequent delays are possible, which is better for security
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Drawbacks of the Coefficient of Variation

At CHES’09, $\sigma/\mu$ was suggested as the efficiency criterion. However, $\sigma$ is not a good measure of uncertainty. Example:

$\sigma$ is larger for $X$, but $X$ is better for the attacker!
Recalling the DPA Complexity

From [Mangard CT-RSA’04]:

\[ T_{DPA} \sim \frac{1}{\rho_{max}^2} \]

In presence of timing disarrangement:

\[ \rho_{max} \sim \hat{p} \]

where \( \hat{p} \) is the maximum of the distribution density.

\[ T_{DPA} \sim \frac{1}{\hat{p}^2} \]

So the key parameter is \( \hat{p} \), not \( \sigma \).
The New Criterion

\[ E = \frac{1}{2\hat{p}\mu}, \quad E \in ]0, 1] \]

\( E = 1 \) when the distribution is uniform, otherwise \( E < 1 \).

**Information-theoretic sense**

Min-entropy:

\[ H_\infty(S) = - \log \hat{p} , \quad H_\infty(S) \leq H(S) \]

where \( H(S) \) is the Shannon entropy.

\[ E = \frac{2^{H_\infty(S)} - 1}{\mu} \]
Outline

1. Random Delays as a Countermeasure
2. Method of CHES’09 and its Limitations
3. Improved Method for Random Delay Generation
4. Correct Efficiency Criterion
5. Practical Evaluation
Practical Evaluation: Implementation

- AES-128 on Atmel ATmega16
- 10 delays per round, 3 dummy rounds at start/end
- almost the same performance overhead for all methods
- no other countermeasures
- CPA attack [Brier et al. CHES’04]
## Practical Evaluation: Results

<table>
<thead>
<tr>
<th></th>
<th>ND</th>
<th>PU</th>
<th>WISTP07</th>
<th>CHES09</th>
<th>CHES10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$, cycles</td>
<td>0</td>
<td>720</td>
<td>860</td>
<td>862</td>
<td>953</td>
</tr>
<tr>
<td>$\hat{p}$</td>
<td>1</td>
<td>0.014</td>
<td>0.009</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>$1/(2\hat{p}\mu)$</td>
<td>–</td>
<td>0.048</td>
<td>0.063</td>
<td>0.145</td>
<td>0.259</td>
</tr>
<tr>
<td>CPA, traces</td>
<td>50</td>
<td>2500</td>
<td>7000</td>
<td>45000</td>
<td>&gt; 150000</td>
</tr>
</tbody>
</table>
Conclusion

Our result

- more secure method for random delay generation
  
  *allows for more frequent but shorter delays*

- correct efficiency criterion
  
  *directly related to the attack complexity and information-theoretically sound*