Analysis and Improvement of the Random Delay Countermeasure of CHES 2009

Jean-Sébastien Coron   Ilya Kizhvatov

UNIVERSITÉ DU LUXEMBOURG

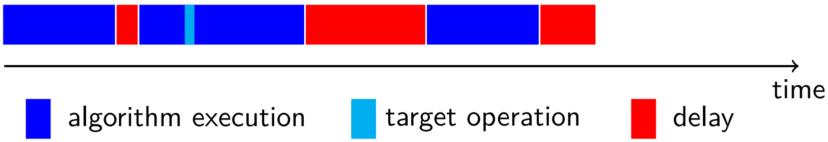
CHES 2010, Santa-Barbara, CA, USA

Outline

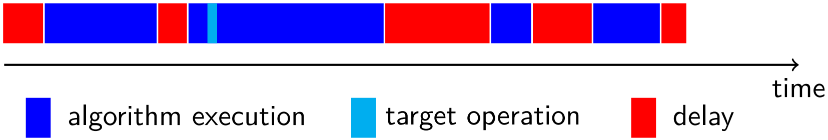
Outline

Random Delays: In Brief

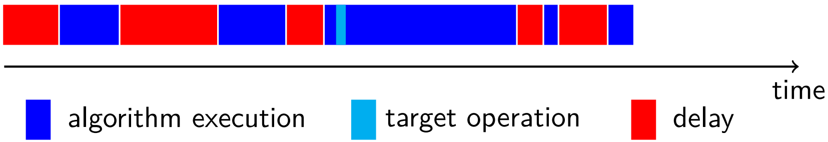
Random Delays: In Brief



Random Delays: In Brief

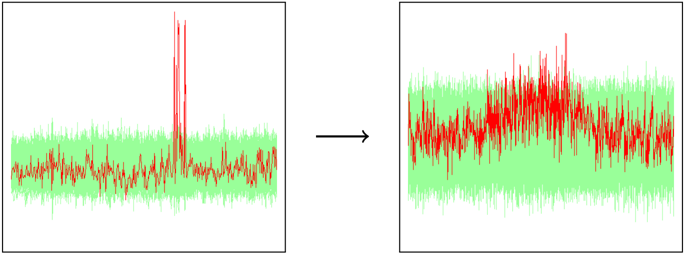


Random Delays: In Brief

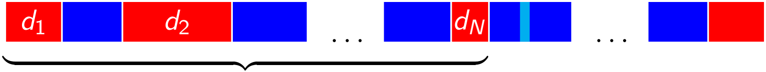


Random Delays: In Brief

Effect in DPA



Random Delays: More Details



$$S_N = \sum_{i=0}^{N} d_i$$

Assumptions

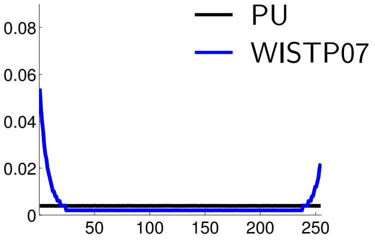
- multiple delays are harder to remove than a single one
- adversary is facing the cumulative sum of N delays

Desired properties of S_N

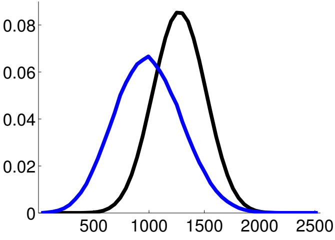
- should increase attacker's **uncertainty**
- **smaller mean** to decrease performance penalty

Methods with Independent Delay Generation

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



1 delay



10 delays

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

Outline

Method of CHES'09: Floating Mean

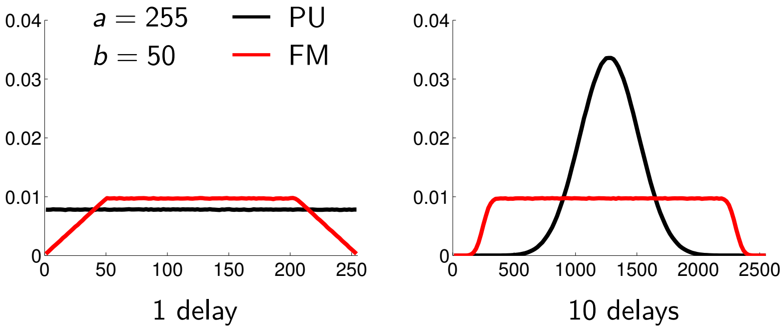
Idea: generate delays non-independently

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

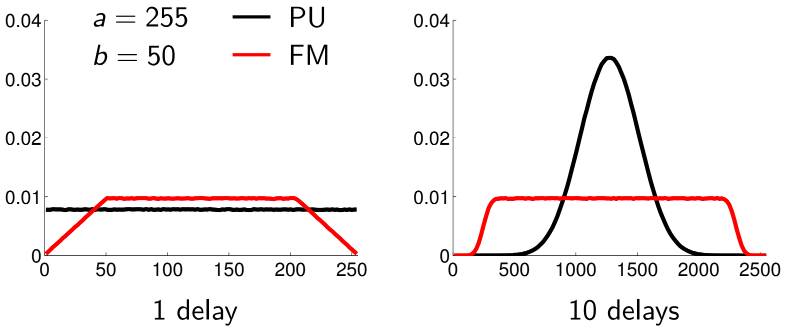
Method of CHES'09: Floating Mean

$$E(S_N) = \frac{Na}{2}, \qquad \mathrm{Var}(S_N) = N^2 \cdot \frac{(a-b+1)^2 - 1}{12} + N \cdot \frac{b^2 + 2b}{12}$$



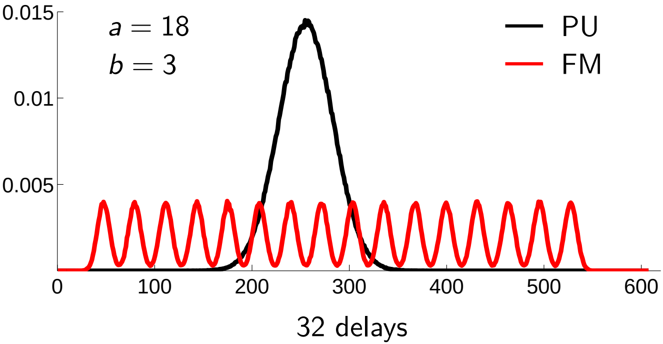
Method of CHES'09: Floating Mean

$$E(S_N) = \frac{Na}{2}\,, \qquad \mathrm{Var}(S_N) = \boxed{N^2} \cdot \frac{(a-b+1)^2-1}{12} + N \cdot \frac{b^2+2b}{12}$$



The Issue with Floating Mean

Using parameters from the practical implementation of CHES'09:



- cogs are not good for security
- σ 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

Improved Floating Mean

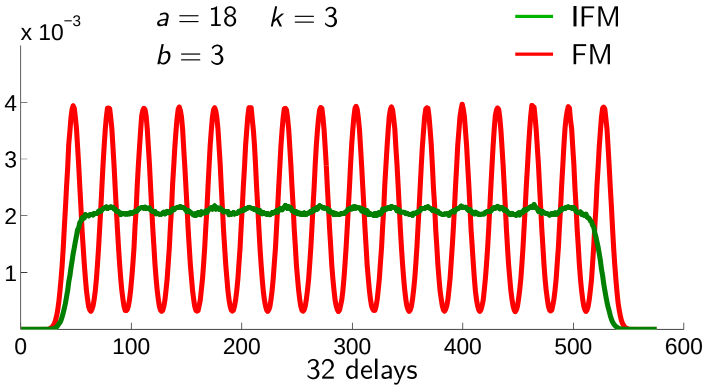
Algorithm

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

$$\mathrm{E}[S_N] = N \cdot \left(\frac{a}{2} - 2^{-k-1}\right) , \qquad \mathrm{Var}(S_N) \simeq N^2 \cdot \frac{(a-b)^2 - 1}{12}$$



Condition on Parameters

Cogs are not visible when

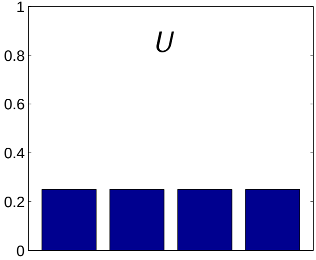
$$b \gg \sqrt{N} \cdot 2^{-k}$$

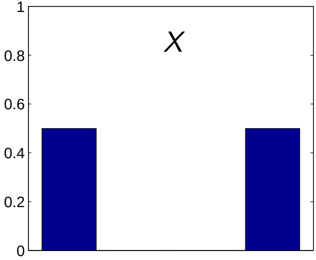
$\Rightarrow$ shorter and more frequent delays are possible, which is better for security

Outline

Drawbacks of the Coefficient of Variation

At CHES'09, σ/μ was suggested as the efficiency criterion.
However, σ is not a good measure of uncertainty. Example:



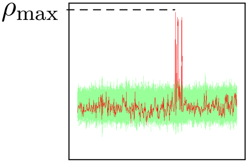


σ is larger for X, but X is better for the attacker!

Recalling the DPA Complexity

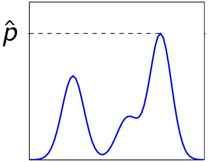
From [Mangard CT-RSA'04]:

$$T_{\mathrm{DPA}} \sim \frac{1}{\rho_{\mathrm{max}}^2}$$



In presence of timing disarrangement:

$$\rho_{\mathrm{max}} \sim \hat{p}$$



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

$$T_{\mathrm{DPA}} \sim \frac{1}{\hat{p}^2}$$

So the key parameter is $\hat{p}$, not σ.

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:

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

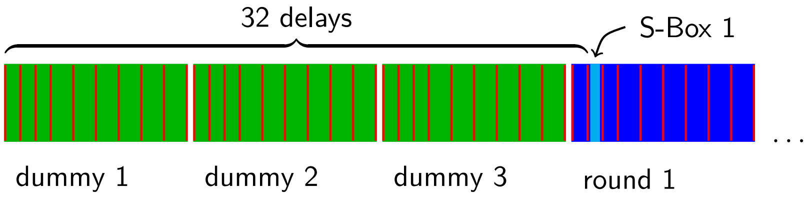
where $H(S)$ is the Shannon entropy.

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

Outline

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

	ND	**PU**	**WISTP07**	**CHES09**	**CHES10**
μ, cycles	0	720	860	862	953
$\hat{p}$	1	0.014	0.009	0.004	0.002
$1/(2\hat{p}\mu)$	–	0.048	0.063	**0.145**	**0.259**
CPA, traces	50	2500	7000	**45000**	$>$ **150000**

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