Attack and Improvement of a Secure S-box Calculation Based on the Fourier Transform

Jean-Sébastien Coron[1], Christophe Giraud[2], Emmanuel Prouff[2], and Matthieu Rivain[1,2]

[1] University of Luxembourg

[2] Oberthur Technologies

August 11, 2008

Outline

Outline

1. Preliminaries
2. S-box Masking Based on the Fourier Transform
3. Differential Power Analysis *vs.* Biased Masking
4. DPA against the FT-Based S-box Masking
5. Improved FT-Based S-box Masking
6. Conclusion

Differential Power Analysis (DPA)

DPA Basics

- **Physical leakage** dependent on **intermediate variables**
- **Sensitive variable** depends on both the input plaintext and on a guessable part of the secret key
- **DPA** exploits the physical leakage on a sensitive variable for key recovery

Differential Power Analysis (DPA)

DPA Basics

- **Physical leakage** dependent on **intermediate variables**
- **Sensitive variable** depends on both the input plaintext and on a guessable part of the secret key
- **DPA** exploits the physical leakage on a sensitive variable for key recovery

DPA Security

Every intermediate variable is independent of any sensitive variable.

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - the key additions
  - the linear transformations
  - the substitution boxes (S-boxes)

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - **the key additions**
  - the linear transformations
  - the substitution boxes (S-boxes)

Key addition

Masked Var.		Mask		
$Z \oplus R$	$\oplus$	R	$=$	Z

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - **the key additions**
  - the linear transformations
  - the substitution boxes (S-boxes)

Key addition

Masked Var.		Mask		
$Z \oplus R \oplus K$	$\oplus$	R	$=$	$Z \oplus K$

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - the key additions
  - **the linear transformations**
  - the substitution boxes (S-boxes)

Linear transformation

Masked Var.		Mask		
$Z \oplus R$	$\oplus$	R	$=$	Z

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - the key additions
  - **the linear transformations**
  - the substitution boxes (S-boxes)

Linear transformation

Masked Var.		Mask		
$L(Z \oplus R)$	$\oplus$	$L(R)$	$=$	$L(Z)$

Masking & S-box protection issue

Masking Countermeasure

- Every sensitive variable Z is masked with a random value R
- **masked variable** $\widetilde{Z} = Z \oplus R$ and **mask** R both independent of Z
- Masked variables and masks processed separately
- Completeness: $Z = \widetilde{Z} \oplus R$

- Masking a block cipher requires the masking of:
  - the key additions
  - the linear transformations
  - **the substitution boxes (S-boxes)**

Substitution box

Issue: From $Z \oplus R$ and R, compute $F(Z) \oplus R'$.
All intermediate var. must be independent of Z.

Outline

S-box Masking Based on the Fourier Transform

- Prouff, Giraud, and Aumonier in CHES 2006 : *Provably Secure S-Box Implementation Based on Fourier Transform*

S-box Masking Based on the Fourier Transform

- Prouff, Giraud, and Aumonier in CHES 2006 : *Provably Secure S-Box Implementation Based on Fourier Transform*
- The Fourier Transform of a $(n \times n)$ S-box F is defined by:

$$\widehat{F}(Z) = \sum_{a \in \mathbb{F}_2^n} F(a)(-1)^{a \cdot Z} .$$

S-box Masking Based on the Fourier Transform

- Prouff, Giraud, and Aumonier in CHES 2006 : *Provably Secure S-Box Implementation Based on Fourier Transform*
- The Fourier Transform of a $(n \times n)$ S-box F is defined by:

$$\widehat{F}(Z) = \sum_{a \in \mathbb{F}_2^n} F(a)(-1)^{a \cdot Z} \ .$$

- It satisfies $\widehat{\widehat{F}} = 2^n F$, that is:

$$F(Z) = \frac{1}{2^n} \widehat{\widehat{F}}(Z) = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z}$$

S-box Masking Based on the Fourier Transform

S-box Masking Based on the Fourier Transform

INPUTS: a masked var. $\widetilde{Z} = Z \oplus R_1$, a mask R_1, a look-up table $\widehat{F}$
OUTPUTS: a masked output $F(Z) \oplus R_3$, a mask R_3

$$F(Z) = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z}$$

S-box Masking Based on the Fourier Transform

S-box Masking Based on the Fourier Transform

INPUTS: a masked var. $\widetilde{Z} = Z \oplus R_1$, a mask R_1, a look-up table $\widehat{F}$
OUTPUTS: a masked output $F(Z) \oplus R_3$, a mask R_3

$$(-1)^{\widetilde{Z} \cdot R_1} F(Z) = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot \widetilde{Z} \oplus R_1 \cdot (a \oplus \widetilde{Z})}$$

S-box Masking Based on the Fourier Transform

S-box Masking Based on the Fourier Transform

INPUTS: a masked var. $\widetilde{Z} = Z \oplus R_1$, a mask R_1, a look-up table $\widehat{F}$
OUTPUTS: a masked output $F(Z) \oplus R_3$, a mask R_3

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) = \frac{1}{2^n} \sum_{a\in\mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)}$$

S-box Masking Based on the Fourier Transform

S-box Masking Based on the Fourier Transform

INPUTS: a masked var. $\widetilde{Z} = Z \oplus R_1$, a mask R_1, a look-up table $\widehat{F}$
OUTPUTS: a masked output $F(Z) \oplus R_3$, a mask R_3

$$(-1)^{(\widetilde{Z} \oplus R_2) \cdot R_1} F(Z) + R_3 \bmod 2^n =$$
$$\frac{1}{2^n}\left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot \widetilde{Z} \oplus R_1 \cdot (a \oplus \widetilde{Z} \oplus R_2)} \bmod 2^{2n}\right)$$

S-box Masking Based on the Fourier Transform

INPUTS: a masked var. $\widetilde{Z} = Z \oplus R_1$, a mask R_1, a look-up table $\widehat{F}$

OUTPUTS: a masked output $F(Z) \oplus R_3$, a mask R_3

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) + R_3 \bmod 2^n =$$

$$\frac{1}{2^n}\left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)} \bmod 2^{2n}\right)$$

Remark

The sum is implemented by a loop on 2^n elements.

$\Rightarrow$ *Of interest for S-boxes with small dimensions (e.g. $n = 4$).*

The Flaw

S-box Masking Based on the Fourier Transform

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) + R_3 = \frac{1}{2^n}\Big(2^n R_3 + R_4 + \sum_{a\in\mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)}\Big)$$

The Flaw

S-box Masking Based on the Fourier Transform

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) + R_3 = \frac{1}{2^n}\left(2^n R_3 + R_4 + \sum_{a\in\mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)}\right)$$

The Flaw

- $a \cdot \widetilde{Z} \oplus R_1 \cdot (\widetilde{Z} \oplus a \oplus R_2) = a \cdot Z \oplus R_1 \cdot (\widetilde{Z} \oplus R_2)$

The Flaw

S-box Masking Based on the Fourier Transform

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) + R_3 = \frac{1}{2^n}\left(2^n R_3 + R_4 + \sum_{a\in\mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)}\right)$$

The Flaw

- $a \cdot \widetilde{Z} \oplus R_1 \cdot (\widetilde{Z} \oplus a \oplus R_2) = a \cdot Z \oplus R_1 \cdot (\widetilde{Z} \oplus R_2)$
- R_1 and $(\widetilde{Z} \oplus R_2)$ are independently and uniformly distributed (iud)

The Flaw

S-box Masking Based on the Fourier Transform

$$(-1)^{(\widetilde{Z}\oplus R_2)\cdot R_1} F(Z) + R_3 = \frac{1}{2^n}\left(2^n R_3 + R_4 + \sum_{a\in\mathbb{F}_2^n} \widehat{F}(a)(-1)^{a\cdot\widetilde{Z}\oplus R_1\cdot(a\oplus\widetilde{Z}\oplus R_2)}\right)$$

The Flaw

- $a\cdot\widetilde{Z}\oplus R_1\cdot(\widetilde{Z}\oplus a\oplus R_2) = a\cdot Z\oplus R_1\cdot(\widetilde{Z}\oplus R_2)$
- R_1 and $(\widetilde{Z}\oplus R_2)$ are independently and uniformly distributed (iud)
- *The scalar product of two iud r. v. $X\cdot Y$ is not a uniform r. v.:*

$$\mathrm{P}\left[X\cdot Y = 0\right] = \frac{1}{2} + \frac{1}{2^{n+1}} .$$

Outline

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} = 0\right] - \mathrm{E}\left[L|b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} = 0\right] - \mathrm{E}\left[L|b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} = 0\right] - \mathrm{E}\left[L|b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- For several executions, measure L and predict $b_k = f(X, k)$.

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} = 0\right] - \mathrm{E}\left[L|b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- For several executions, measure L and predict $b_k = f(X, k)$.
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L|b_k = 0\right] - \widehat{\mathrm{E}}\left[L|b_k = 1\right]$

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L | b_{k^*} = 0\right] - \mathrm{E}\left[L | b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- For several executions, measure L and predict $b_k = f(X, k)$.
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L | b_k = 0\right] - \widehat{\mathrm{E}}\left[L | b_k = 1\right]$
  - If $k = k^*$ then $\mathrm{P}\left[b_k = b_{k^*}\right] = 1$ and $\Delta_k \to \Delta$

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} = 0\right] - \mathrm{E}\left[L|b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- For several executions, measure L and predict $b_k = f(X, k)$.
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L|b_k = 0\right] - \widehat{\mathrm{E}}\left[L|b_k = 1\right]$
  - If $k = k^*$ then $\mathrm{P}\left[b_k = b_{k^*}\right] = 1$ and $\Delta_k \to \Delta$
  - If $k \neq k^*$ then $\mathrm{P}\left[b_k = b_{k^*}\right] = \alpha < 1$ and $\Delta_k \to (1 - 2\alpha)\Delta$

Oberthur Technologies

DPA Principle

- Let $b_{k^*} = f(X, k^*)$ be a bit of the computation, where
  - X is a public variable (uniformly distributed)
  - k^* is a guessable part of the secret key
- Let L be the leakage on b_{k^*}

DPA Assumption

$$\mathrm{E}\left[L | b_{k^*} = 0\right] - \mathrm{E}\left[L | b_{k^*} = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- For several executions, measure L and predict $b_k = f(X, k)$.
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L | b_k = 0\right] - \widehat{\mathrm{E}}\left[L | b_k = 1\right]$
  - If $k = k^*$ then $\mathrm{P}\left[b_k = b_{k^*}\right] = 1$ and $\Delta_k \to \Delta$
  - If $k \neq k^*$ then $\mathrm{P}\left[b_k = b_{k^*}\right] = \alpha < 1$ and $\Delta_k \to (1 - 2\alpha)\Delta$
- Assuming $\alpha > 0$ we have $|(1 - 2\alpha)\Delta| < |\Delta|$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}[R = 0] = \frac{1}{2} + \varepsilon$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}[R = 0] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}[L | b_{k^*} \oplus R = 0] - \mathrm{E}[L | b_{k^*} \oplus R = 1] = \Delta \neq 0$$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}\left[R = 0\right] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} \oplus R = 0\right] - \mathrm{E}\left[L|b_{k^*} \oplus R = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L|b_k = 0\right] - \widehat{\mathrm{E}}\left[L|b_k = 1\right]$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}\left[R = 0\right] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}\left[L|b_{k^*} \oplus R = 0\right] - \mathrm{E}\left[L|b_{k^*} \oplus R = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L|b_k = 0\right] - \widehat{\mathrm{E}}\left[L|b_k = 1\right]$
- If $k = k^*$ then $\mathrm{P}\left[b_k = b_{k^*} \oplus R\right] = \frac{1}{2} + \varepsilon$, and
  - $\Delta_k \to \left(\frac{1}{2} + \varepsilon\right)\Delta + \left(\frac{1}{2} - \varepsilon\right)(-\Delta)$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}\left[R = 0\right] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}\left[L | b_{k^*} \oplus R = 0\right] - \mathrm{E}\left[L | b_{k^*} \oplus R = 1\right] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}\left[L | b_k = 0\right] - \widehat{\mathrm{E}}\left[L | b_k = 1\right]$
- If $k = k^*$ then $\mathrm{P}\left[b_k = b_{k^*} \oplus R\right] = \frac{1}{2} + \varepsilon$, and
  - $\Delta_k \rightarrow \left(\frac{1}{2} + \varepsilon\right)\Delta + \left(\frac{1}{2} - \varepsilon\right)(-\Delta) = 2\varepsilon\Delta$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}[R = 0] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}[L | b_{k^*} \oplus R = 0] - \mathrm{E}[L | b_{k^*} \oplus R = 1] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}[L | b_k = 0] - \widehat{\mathrm{E}}[L | b_k = 1]$
- If $k = k^*$ then $\mathrm{P}[b_k = b_{k^*} \oplus R] = \frac{1}{2} + \varepsilon$, and
  - $\Delta_k \to \left(\frac{1}{2} + \varepsilon\right)\Delta + \left(\frac{1}{2} - \varepsilon\right)(-\Delta) = 2\varepsilon\Delta$
- If $k \neq k^*$ then $\Delta_k \to 2\varepsilon(1 - 2\alpha)\Delta$

DPA *vs.* Biased Masking

- The leakage L depends on a masked bit $b_{k^*} \oplus R$
- The mask is biased: $\mathrm{P}[R = 0] = \frac{1}{2} + \varepsilon$

DPA Assumption

$$\mathrm{E}[L|b_{k^*} \oplus R = 0] - \mathrm{E}[L|b_{k^*} \oplus R = 1] = \Delta \neq 0$$

DPA Attack

- Make a guess $k \stackrel{?}{=} k^*$
- Compute the difference of means: $\Delta_k = \widehat{\mathrm{E}}[L|b_k = 0] - \widehat{\mathrm{E}}[L|b_k = 1]$
- If $k = k^*$ then $\mathrm{P}[b_k = b_{k^*} \oplus R] = \frac{1}{2} + \varepsilon$, and
  - $\Delta_k \to (\frac{1}{2} + \varepsilon)\Delta + (\frac{1}{2} - \varepsilon)(-\Delta) = 2\varepsilon\Delta$
- If $k \neq k^*$ then $\Delta_k \to 2\varepsilon(1 - 2\alpha)\Delta$
- The convergence requires about $(\frac{1}{2\varepsilon})^2$ times more leakage measurements

Outline

DPA against the FT-Based S-box Masking

- Targeted bit: $a \cdot Z \oplus R_1 \cdot (\widetilde{Z} \oplus R_2)$
  - Z: sensitive n-bit S-box input
  - a: loop index
  - $R_1 \cdot (\widetilde{Z} \oplus R_2)$: biased mask

DPA against the FT-Based S-box Masking

- Targeted bit: $a \cdot Z \oplus R_1 \cdot (\widetilde{Z} \oplus R_2)$
  - Z: sensitive n-bit S-box input
  - a: loop index
  - $R_1 \cdot (\widetilde{Z} \oplus R_2)$: biased mask
- Mask bias: $\varepsilon = \frac{1}{2^{n+1}}$
  - Number of required measurements multiply by $(\frac{1}{2\varepsilon})^2 = 2^{2n}$
  - If $n = 4$ then $(\frac{1}{2\varepsilon})^2 = 256$

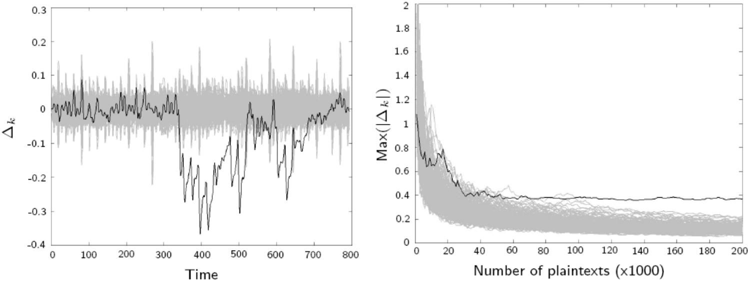
DPA against the FT-Based S-box Masking
Practical Experiment

- Masked AES implementation
- S-box implemented with the *composite field method*
- F is defined as : $F(x) = \begin{cases} x^{-1} & \text{if } x \in GF(16)\backslash\{0\} \\ 0 & \text{if } x = 0 \end{cases}$

DPA against the FT-Based S-box Masking

Practical Experiment

- Masked AES implementation
- S-box implemented with the *composite field method*
- F is defined as : $F(x) = \begin{cases} x^{-1} & \text{if } x \in GF(16)\backslash\{0\} \\ 0 & \text{if } x = 0 \end{cases}$



(a) Value of Δ_k. (b) Convergence of $|\Delta_k|$.

Outline

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$
\begin{aligned}
&(-1)^{R_2} F(Z) + R_3 \bmod 2^n \\
&\quad = \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,
\end{aligned}
$$

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$
\begin{aligned}
&(-1)^{R_2} F(Z) + R_3 \bmod 2^n \\
&\quad = \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,
\end{aligned}
$$

- For every a:

$$
Tmp \leftarrow a \cdot \widetilde{Z} \qquad [Tmp = a \cdot \widetilde{Z}]
$$

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$(-1)^{R_2} F(Z) + R_3 \bmod 2^n$$
$$= \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,$$

- For every a:

$$\begin{array}{ll} Tmp \leftarrow a \cdot \widetilde{Z} & [Tmp = a \cdot \widetilde{Z}] \\ Tmp \leftarrow Tmp \oplus R_2 & [Tmp = a \cdot \widetilde{Z} \oplus R_2] \end{array}$$

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$(-1)^{R_2} F(Z) + R_3 \bmod 2^n$$
$$= \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,$$

- For every a:

$$\begin{array}{ll} Tmp \leftarrow a \cdot \widetilde{Z} & [Tmp = a \cdot \widetilde{Z}] \\ Tmp \leftarrow Tmp \oplus R_2 & [Tmp = a \cdot \widetilde{Z} \oplus R_2] \\ Tmp \leftarrow Tmp \oplus a \cdot R_1 & [Tmp = a \cdot Z \oplus R_2] \end{array}$$

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$(-1)^{R_2} F(Z) + R_3 \bmod 2^n = \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,$$

- For every a:

$$\begin{array}{ll} Tmp \leftarrow a \cdot \widetilde{Z} & [Tmp = a \cdot \widetilde{Z}] \\ Tmp \leftarrow Tmp \oplus R_2 & [Tmp = a \cdot \widetilde{Z} \oplus R_2] \\ Tmp \leftarrow Tmp \oplus a \cdot R_1 & [Tmp = a \cdot Z \oplus R_2] \end{array}$$

- DPA Security: exponent masked by R_2, sum masked by (R_3, R_4)

Improved FT-Based S-box Masking

- The exponent is masked with one random bit R_2

$$(-1)^{R_2} F(Z) + R_3 \bmod 2^n = \left\lfloor \frac{1}{2^n} \left(2^n R_3 + R_4 + \sum_{a \in \mathbb{F}_2^n} \widehat{F}(a)(-1)^{a \cdot Z \oplus R_2} \bmod 2^{2n} \right) \right\rfloor ,$$

- For every a:

$$\begin{array}{ll} Tmp \leftarrow a \cdot \widetilde{Z} & [Tmp = a \cdot \widetilde{Z}] \\ Tmp \leftarrow Tmp \oplus R_2 & [Tmp = a \cdot \widetilde{Z} \oplus R_2] \\ Tmp \leftarrow Tmp \oplus a \cdot R_1 & [Tmp = a \cdot Z \oplus R_2] \end{array}$$

- DPA Security: exponent masked by R_2, sum masked by (R_3, R_4)
- Efficiency: 2^{n+1} look-ups avoided

Outline

Conclusion

- The FT-based DPA countermeasure of CHES 2006 has a flaw
- The flaw makes an efficient DPA attack possible
- Our attack has been practically validated
- We propose an improved version of the countermeasure
  - provably secure against DPA
  - more efficient than the original countermeasure