Uniform First-Order Threshold Implementations

Tim Beyne, Begül Bilgin

COSIC / ESAT, KULeuven and iMinds, Belgium

August 10, 2016

Threshold Implementations
Introduction

- Countermeasure against side-channel attacks
  - First-order attacks: provably secure
  - Higher-order attacks: not in this paper
- Based on secret sharing and multi-party computation
  - Input is split into random *shares*: *sharing*
  - Function is split into shares: *realization*
- Implementation cost increases with number of shares
  - More gates
  - More randomness (sometimes)

Threshold Implementations
Definitions

- $x \in \mathbb{F}_2$ is split into random shares $x_1, \ldots, x_s$ ("sharing")
- $\mathbf{x} = (x_1, \ldots, x_s)$ is a *correct* sharing:

$$x = \bigoplus_{i=1}^{s} x_i$$

- A sharing is uniformly generated if, for all x, every correct sharing $\mathbf{x}$ is equally likely

Threshold Implementations

Definitions

- Unshared Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$(x^1, \ldots, x^n) \mapsto f(x^1, \ldots, x^n)$$

- Realization $\mathbf{f} = (f_1, f_2, \ldots, f_{s_{\text{out}}})$ with $f_i : {\mathbb{F}_2^n}^{s_{\text{in}}} \to \mathbb{F}_2$
- Correctness

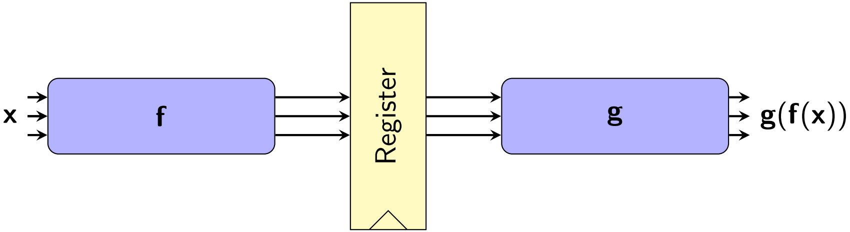
$$f(x^1, \ldots, x^n) = \bigoplus_{i=1}^{s_{\text{out}}} f_i(\mathbf{x}^1, \ldots, \mathbf{x}^n)$$

- Noncompleteness: each f_i is independent of x_i^j $(\forall j, 1 \leq j \leq n)$
- Vectorial functions: repeat for each coordinate function

Threshold Implementations

Security guarantees

- Output share does not reveal anything about a *uniformly shared* input
- Output $\mathbf{f}$ must be uniform when cascading functions
  = "uniformity property"
- $\mathbf{g} \circ \mathbf{f}$ is secure against *first-order* attacks if $\mathbf{f}(\mathbf{x})$ is uniformly generated



Threshold Implementations

Example

- $f(x^1, x^2) = x^1 x^2 \; (x^i = x_1^i \oplus x_2^i \oplus x_3^i)$
- $f_1 \oplus f_2 \oplus f_3 = (x_1^1 \oplus x_2^1 \oplus x_3^1) \cdot (x_1^2 \oplus x_2^2 \oplus x_3^2)$

$$
\begin{aligned}
f_1 &= x_2^1 x_2^2 \oplus x_2^1 x_3^2 \oplus x_3^1 x_2^2 \\
f_2 &= x_1^1 x_3^2 \oplus x_3^1 x_1^2 \oplus x_3^1 x_3^2 \\
f_3 &= x_1^1 x_1^2 \oplus x_1^1 x_2^2 \oplus x_2^1 x_1^2
\end{aligned}
$$

				(f_1, f_2, f_3)				
(x^1, x^2)	000	001	010	011	100	101	110	111
00	**7**	0	0	**3**	0	**3**	**3**	0
01	**7**	0	0	**3**	0	**3**	**3**	0
10	**7**	0	0	**3**	0	**3**	**3**	0
11	0	5	5	0	5	0	0	1

Threshold Implementations
Uniformity table

- The uniformity table $\mathcal{U}$ has elements $\mathcal{U}_{x,\mathbf{y}}$
- A realization is uniform iff $\forall x, \mathbf{y}$:

$$\mathcal{U}_{x,\mathbf{y}} = 2^{n(s_{\text{in}}-1)-m(s_{\text{out}}-1)} \text{ or } 0$$

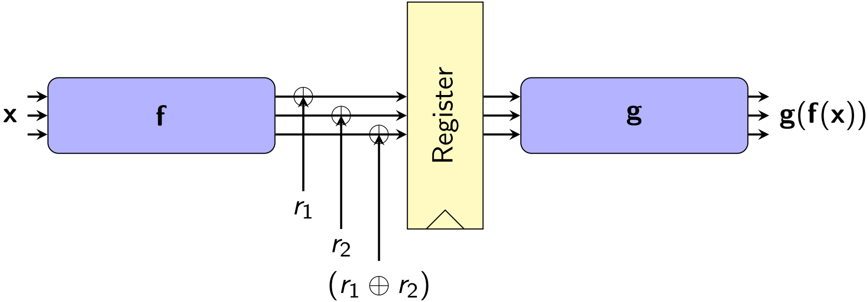
(with m the number of output bits)

	(f_1, f_2, f_3)							
(x^1, x^2)	000	011	101	110	001	010	100	111
00	4	0	0	4	0	4	4	0
01	4	0	0	4	0	4	4	0
10	4	0	0	4	0	4	4	0
11	0	4	4	0	4	0	0	4

Threshold Implementations

Solutions for Uniformity: Remasking

- Adding new randomness ("remasking")
- Randomness is not free
- Example: Keccak-f[1600] with 3 shares
  - 10 bits of randomness per S-box evaluation
  - 24 rounds, 320 S-box evaluations per round



Threshold Implementations

Solutions for Uniformity: Correction Terms

- Adding “correction terms” (CTs) to achieve uniformity
- Add the same term to two output shares

Threshold Implementations

Solutions for Uniformity: Correction Terms

- $f(x^1, x^2) = x^1 x^2$ $(x^i = x_1^i \oplus x_2^i \oplus x_3^i)$
- $f_1 \oplus f_2 \oplus f_3 = (x_1^1 \oplus x_2^1 \oplus x_3^1) \cdot (x_1^2 \oplus x_2^2 \oplus x_3^2)$

$$
\begin{aligned}
f_1 &= x_2^1 x_2^2 \oplus x_2^1 x_3^2 \oplus x_3^1 x_2^2 \color{red}{\oplus x_3^1 \oplus x_3^2} \\
f_2 &= x_1^1 x_3^2 \oplus x_3^1 x_1^2 \oplus x_3^1 x_3^2 \color{red}{\oplus x_3^1 \oplus x_3^2} \\
f_3 &= x_1^1 x_1^2 \oplus x_1^1 x_2^2 \oplus x_2^1 x_1^2
\end{aligned}
$$

	(f_1, f_2, f_3)							
(x^1, x^2)	000	011	101	110	001	010	100	111
00	**5**	0	0	**1**	0	**5**	**5**	0
01	**5**	0	0	**1**	0	**5**	**5**	0
10	**5**	0	0	**1**	0	**5**	**5**	0
11	0	3	3	0	7	0	0	3

Threshold Implementations

Solutions for Uniformity: Correction Terms

- Difficult due to the size of the search space
  (4 bit S-box: $(2^{30})^4$ with linear and quadratic CTs)
- Not always possible (more shares might be required)
  e.g. no known 3-share uniform realization of Keccak-$f\,[b]$

Threshold Implementations

Solutions for Uniformity: Partial Uniformity

- Combination of remasking and correction terms
- If a subset of the output shares is uniform, only remask the others
- Requires less randomness than remasking
  e.g. Keccak-f[1600]: 4 bits / S-box (compare with 10)
- Easier than finding a completely uniform realization

Uniform Not uniform

$$\overbrace{x_1^1 \quad x_2^1 \quad x_3^1}^{x^1} \qquad \overbrace{x_1^2 \quad x_2^2 \quad x_3^2}^{x^2}$$

$$\begin{array}{ccc} \oplus & \oplus & \oplus \\ r_1 & r_2 & (r_1 \oplus r_2) \end{array}$$

Threshold Implementations

Solutions for Uniformity: Partial Uniformity

- Find uniform realizations for each coordinate function of $\mathbf{f}$ by iterating over all CTs
- For $l = 2 \ldots m$, check which l-combinations are uniform
- Problems to solve
  - **Checking uniformity is slow**
  - Search space of correction terms is large

Checking Uniformity
Approach

- Here: Boolean functions (one unshared output bit)
- Naive method: compute the uniformity table (worst-case: $2^{ns_{\text{in}}}$ evaluations of the realization)
- Uniformity table is not random

Checking Uniformity

Restrictions on the Uniformity Table

- The entries of any row of $\mathcal{U}$ are related by *the same* linear equations
- For $s_{\text{out}} = 3$ we have as many equations as unknowns
  - System of equations has a unique solution
  - Any row completely determines $\mathcal{U}$
- Only one row must be checked to check uniformity
- Complexity reduced by factor 2^n
- It also follows that

$$(f_1, f_2, f_3) \text{ is uniform } \iff f_1, f_2, f_3 \text{ are balanced}$$

- $s_{\text{out}} \geq 4$
  - Multiple rows necessary
  - More complicated restrictions on the uniformity table

Threshold Implementations

Solutions for Uniformity: Partial Uniformity

- Find uniform realizations for each coordinate function of $\mathbf{f}$ by iterating over all CTs
- For $l = 2 \ldots m$, check which l-combinations are uniform
- Problems to solve
  - Checking uniformity is slow
  - **Search space of correction terms is large**

Correction Terms

Linear Correction Terms

COSIC

- Walsh-Hadamard transform $\mathcal{W}_{f_i}$ of $f_i : \mathbb{F}_2^{n\,(s_{\text{in}}-1)} \to \mathbb{F}_2$
- $f_i(\mathbf{x}) \oplus \mathbf{a} \cdot \mathbf{x}$ is balanced if and only if $\mathcal{W}_{f_i}(\mathbf{a}) = 0$
- $\mathcal{W}_{f_i}$ can be computed in $O(n(s_{in} - 1)2^{(s_{in}-1)n})$ operations
- $(f_1 \oplus \mathbf{a} \cdot \mathbf{x}, f_2 \oplus \mathbf{b} \cdot \mathbf{x}, f_3 \oplus (\mathbf{a} \oplus \mathbf{b}) \cdot \mathbf{x})$ is uniform if and only if

$$\mathcal{W}_{f_1}(\mathbf{a}) = 0 \text{ with } a_1^i = 0$$
$$\mathcal{W}_{f_2}(\mathbf{b}) = 0 \text{ with } b_2^i = 0$$
$$\mathcal{W}_{f_3}(\mathbf{a} \oplus \mathbf{b}) = 0 \text{ with } a_3^i = b_3^i$$

- Necessary but not sufficient for $s_{\text{out}} > 3$

Correction Terms
Linear Correction Terms

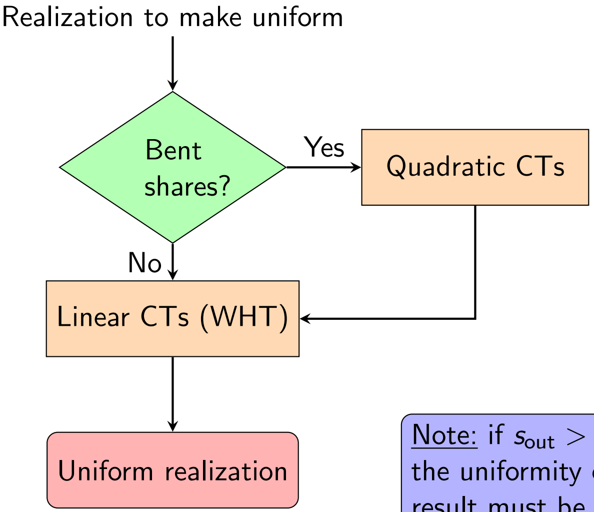
- For a bent function f_i:

$$\forall a \in \mathbb{F}_2^{n(s_{in}-1)} : \mathcal{W}_{f_i}(\mathbf{a}) \neq 0$$

- Impossible to find linear corrections
- Avoid bent functions by using nonlinear correction terms
- e.g. $\mathbb{F}_4$-multiplier used in some AES implementations

Finding Uniform Realizations

Overview for quadratic Boolean functions



Note: if $s_{out} > 3$, the uniformity of the result must be checked

Correction Terms

Quadratic Correction Terms

- Systematic method to avoid bent components for quadratic Boolean functions
- Matrix M_i of the bilinear form of each share f_i
  - Correctness: $\sum_{i=1}^{s_{\text{out}}} M_i = M$
    (M is a block-matrix with $s_{\text{in}} \times s_{\text{in}}$ blocks with values from the matrix of the bilinear form of f)
  - Non-bent: $\operatorname{rank}(M_i) < n(s_{\text{in}} - 1)$.
- Find s_{out} matrices M_i such that both conditions are satisfied

Correction Terms
Quadratic Correction Terms

- There is an invertible T such that $M = TNT^T$ with

$$N = \begin{pmatrix} \begin{smallmatrix} 0 & J \\ J & 0 \end{smallmatrix} & & & \\ & \begin{smallmatrix} 0 & J \\ J & 0 \end{smallmatrix} & & \\ & & \ddots & \\ & & & 0 \end{pmatrix} \text{ with } J = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \in \mathbb{F}_2^{s_{\text{in}} \times s_{\text{in}}}$$

- It is easier to find N_i such that $N = \sum_{i=1}^{s_{\text{out}}} N_i$ with $\operatorname{rank}(N_i) < n(s_{\text{in}} - 1)$
- Let $M_i = TN_iT^T$ (T preserves rank and non-completeness)

Correction Terms

Quadratic Correction Terms

$$N = \begin{pmatrix} 0 & J & & & & \\ J & 0 & & & & \\ & & 0 & J & & \\ & & J & 0 & & \\ & & & & 0 & J \\ & & & & J & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & J_1 & & & & \\ J_1 & 0 & & & & \\ & & 0 & J_1' & & \\ & & J_1' & 0 & & \\ & & & & 0 & J_1'' \\ & & & & J_1'' & 0 \end{pmatrix} + \begin{pmatrix} 0 & J_2 & & & & \\ J_2 & 0 & & & & \\ & & 0 & J_2' & & \\ & & J_2' & 0 & & \\ & & & & 0 & J_2'' \\ & & & & J_2'' & 0 \end{pmatrix} + \begin{pmatrix} 0 & J_3 & & & & \\ J_3 & 0 & & & & \\ & & 0 & J_3' & & \\ & & J_3' & 0 & & \\ & & & & 0 & J_3'' \\ & & & & J_3'' & 0 \end{pmatrix}$$

with

$$J_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, J_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, J_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$J_1' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, J_2' = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, J_3' = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$J_1'' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, J_2'' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, J_3'' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Conclusion

- Theoretical results on the uniformity property
- Efficient method to check uniformity
- Systematic search method for
  - Linear correction terms
  - Quadratic correction terms
- Uniform realizations for most quadratic Boolean functions with only 3 shares
- Specific examples: 50% randomness reduction for
  - $\mathbb{F}_4$-multiplier used in some AES implementations
  - "Problematic" $\mathcal{Q}^4_{300}$ 4-bit permutations
- Future applications: any quadratic function (higher-degree functions can be decomposed first)

COSIC

Thank you for your attention.

Questions?