Higher-order Masking and Shuffling for Software Implementations of Block Ciphers

Matthieu Rivain, Emmanuel Prouff and Julien Doget

Oberthur Technologies, University of Luxembourg, UCL and Paris 8

SCA and Software countermeasures

Attacks

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].
- d^{th} order SCA use d different times per trace [Messerges00].

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].
- d^{th} order SCA use d different times per trace [Messerges00].

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].
- d^{th} order SCA use d different times per trace [Messerges00].

Software Countermeasures (CM) against d^{th} order SCA

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].
- d^{th} order SCA use d different times per trace [Messerges00].

Software Countermeasures (CM) against d^{th} order SCA

- Masking [ChariJultaRaoRohatgi99,GoubinPatarin99].

SCA and Software countermeasures

Attacks

- Algorithm Processing leaks information about the manipulated data
- Side Channel Analyses (SCA) exploit this leakage:
  CPA [BrierClavierOlivier04],
  MIA [GierlichsBatinaTuylsPreneel08],
  Template Attacks [ChariRaoRohatgi02].
- d^{th} order SCA use d different times per trace [Messerges00].

Software Countermeasures (CM) against d^{th} order SCA

- Masking [ChariJultaRaoRohatgi99,GoubinPatarin99].
- Shuffling [HerbstOswaldMangard06].

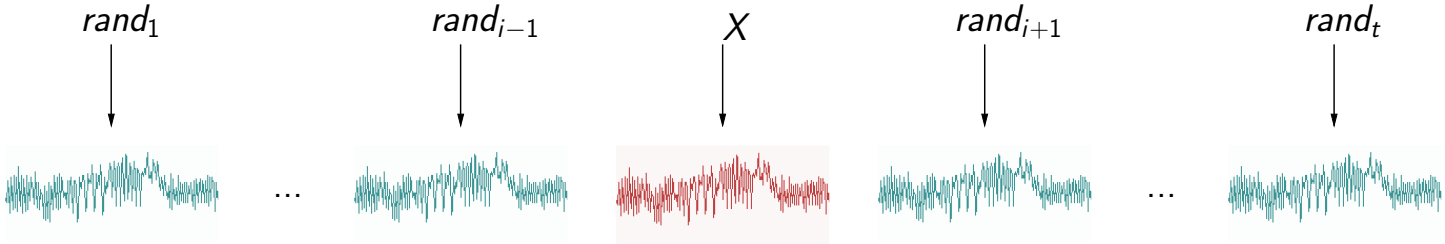
Shuffling Method

Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.

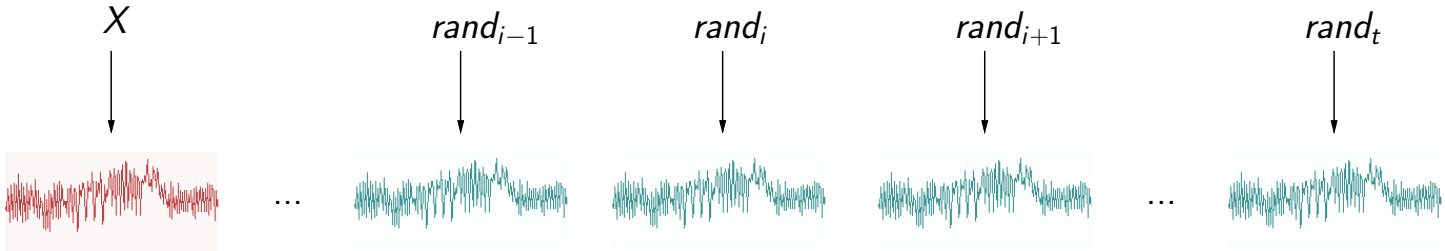
Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



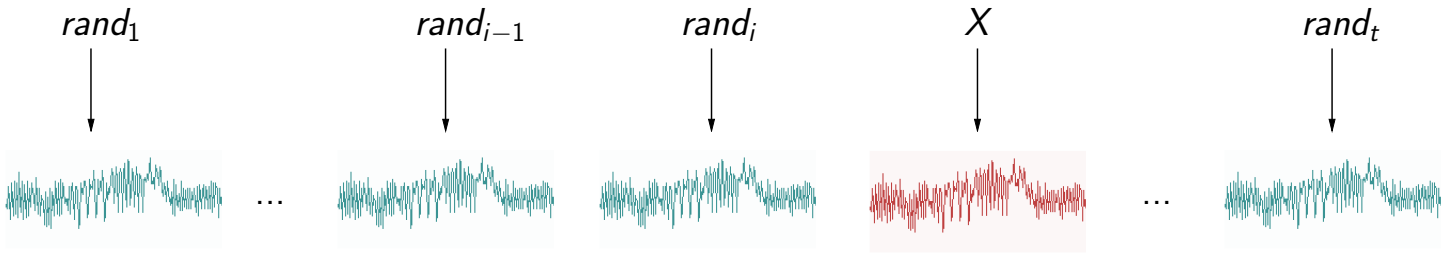
Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



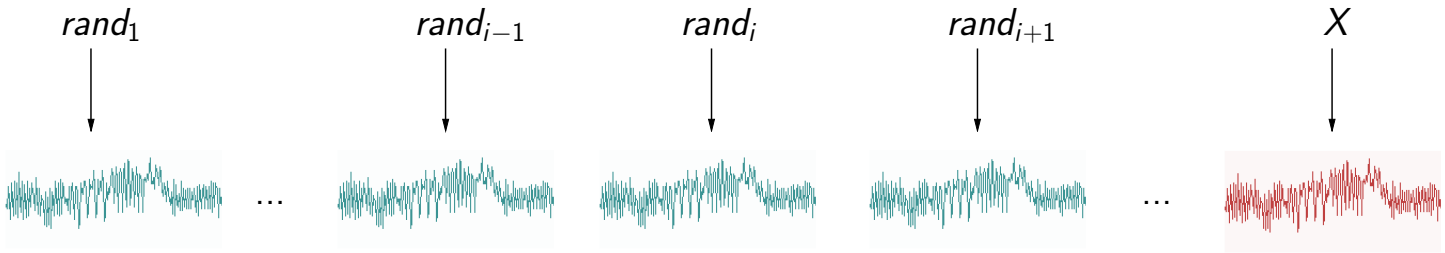
Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



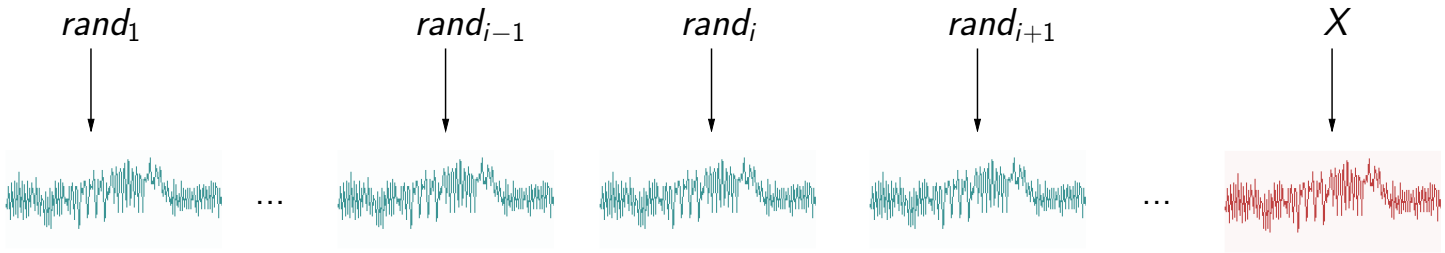
Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



Shuffling Method

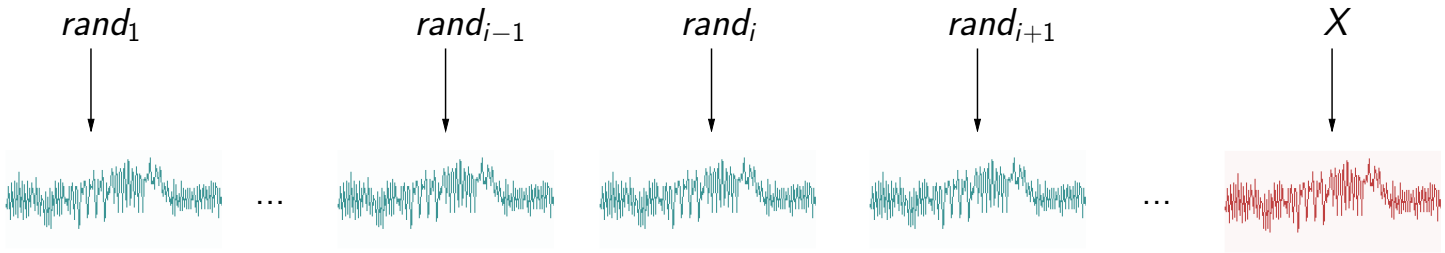
- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



- Impact: decreases the SNR of the instantaneous leakage on X by a factor of t

Shuffling Method

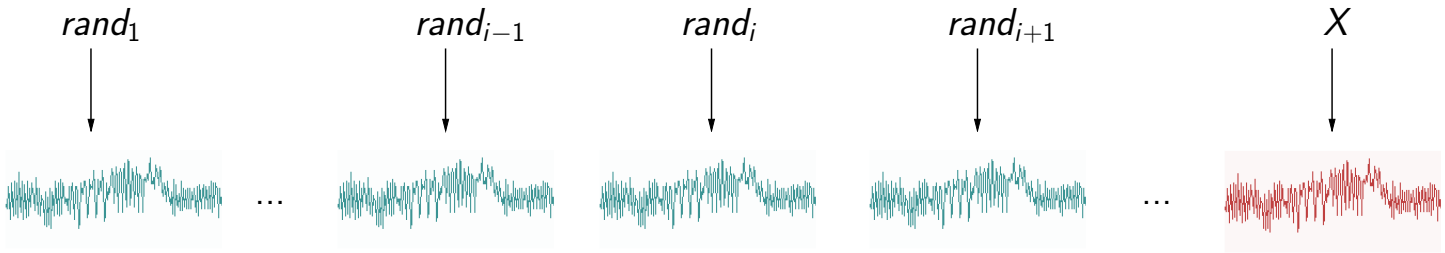
- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



- Impact: decreases the SNR of the instantaneous leakage on X by a factor of t
- Asset: can be straightforwardly adapted to protect any operation Op on X.

Shuffling Method

- Core Idea: spread the sensitive signal related to X over t different signals $S_1, \ldots, S_t$ leaking at different times.
- Select an index at random:



- Impact: decreases the SNR of the instantaneous leakage on X by a factor of t
- Asset: can be straightforwardly adapted to protect any operation Op on X.
- Issue: t must be very large to have satisfying security.

Masking Method

Masking Method

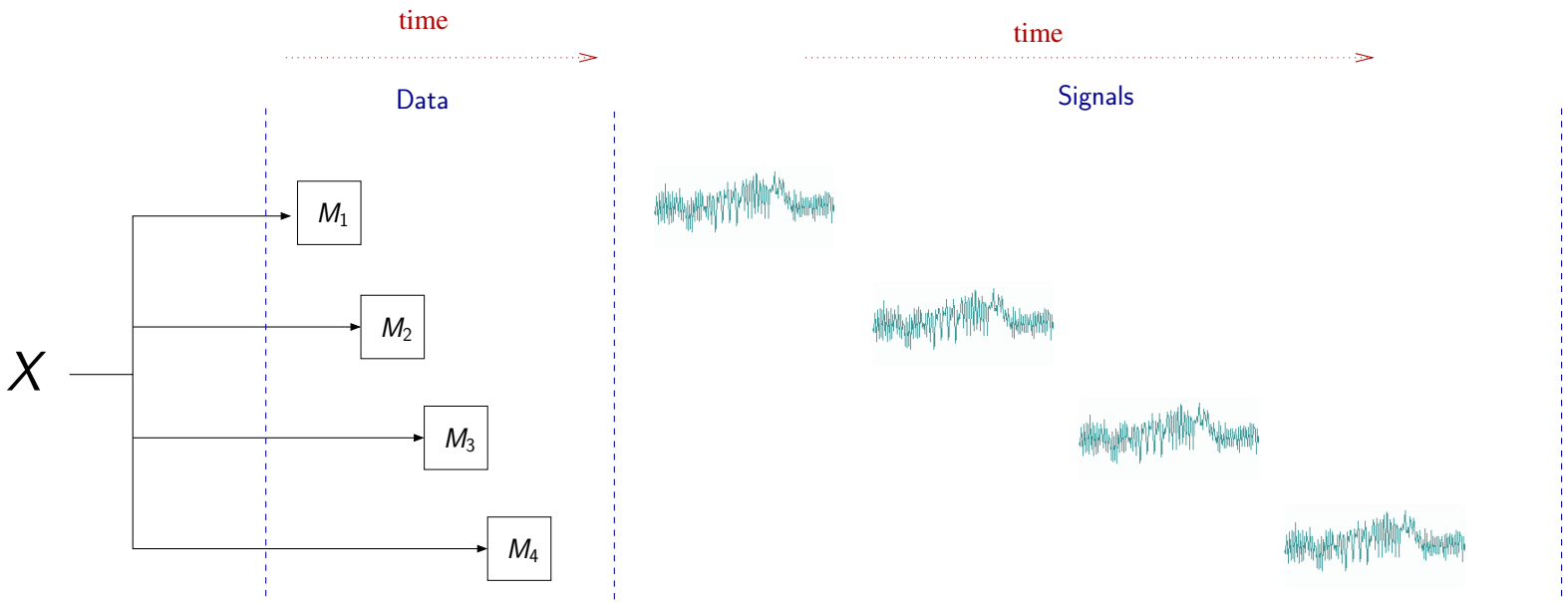
- Core idea: randomly split X into $d+1$ shares M_0,..., M_d s.t

$$M_0 \oplus \cdots \oplus M_d = X \ .$$

Masking Method

- Core idea: randomly split X into $d+1$ shares M_0,..., M_d s.t

$$M_0 \oplus \cdots \oplus M_d = X \ .$$



Masking Method

- Core idea: randomly split X into $d+1$ shares M_0,..., M_d s.t

$$M_0 \oplus \cdots \oplus M_d = X \ .$$

- Impact: complexity of d^{th} order SCA grows exponentially with d [CJRR99].

Masking Method

- Core idea: randomly split X into $d+1$ shares M_0,..., M_d s.t

$$M_0 \oplus \cdots \oplus M_d = X \ .$$

- Impact: complexity of d^{th} order SCA grows exponentially with d [CJRR99].
- Asset: dealing with the propagation of the masks when performing $Op(X)$ is easy when Op is linear.

Masking Method

- Core idea: randomly split X into $d+1$ shares M_0,..., M_d s.t

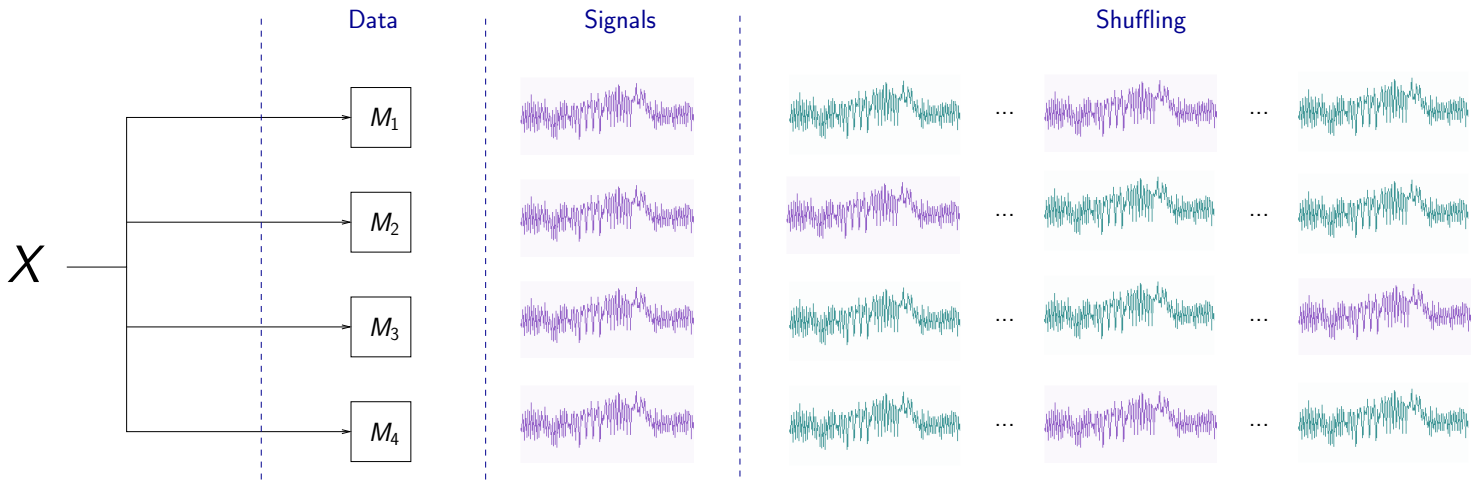
$$M_0 \oplus \cdots \oplus M_d = X \ .$$

- Impact: complexity of d^{th} order SCA grows exponentially with d [CJRR99].
- Asset: dealing with the propagation of the masks when performing $Op(X)$ is easy when Op is linear.
- Issue: even for small d, dealing with the mask propagation is an issue when $Op =$ S-box.
  - Costly solutions exist only for $d \leq 3$ [SchramPaar06,RivainDottaxProuff08b].

Masking-and-Shuffling Method

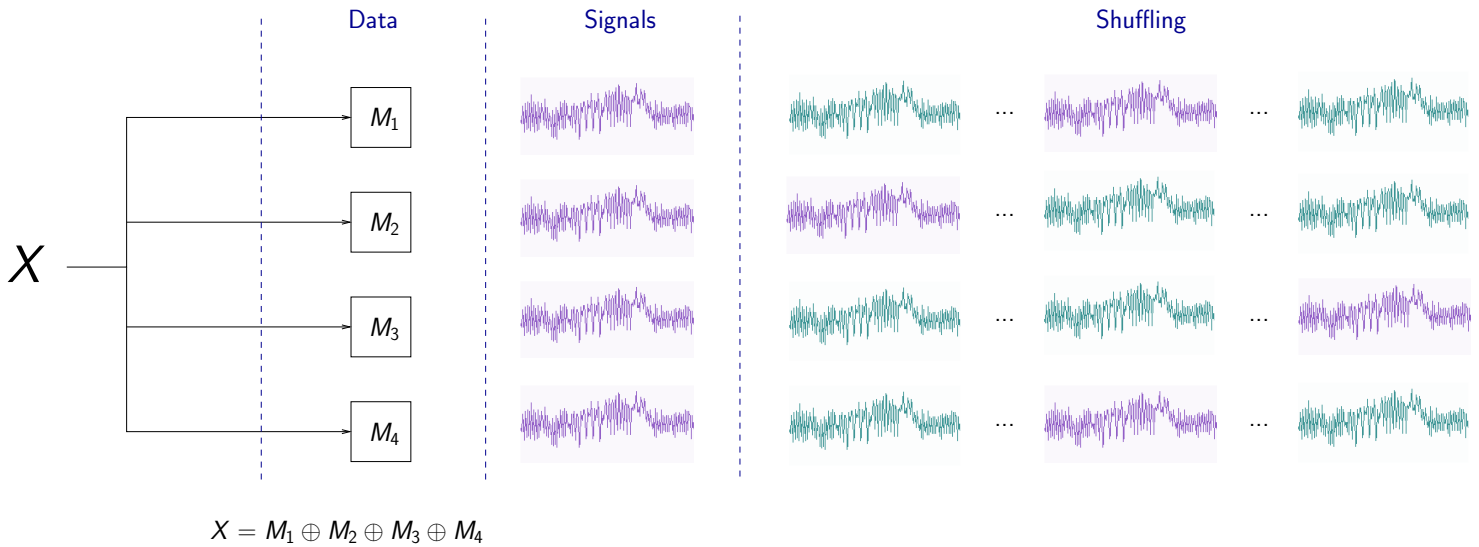
- Core Idea: combine Masking and Shuffling.
- First Proposal: combine 1st-order masking with shuffling [HOM06,TillichHerbstMangard07].
- Analyses in [THM07] and [TillichHerbst08] show that the resulting security is not good.
- Possible Improvement: involve higher-order masking [this paper]

Masking-and-Shuffling Method - Figure



$X = M_1 \oplus M_2 \oplus M_3 \oplus M_4$

Masking-and-Shuffling Method - Figure



Raises two issues

1. How to combine higher-order masking with shuffling?
2. How to quantify the security of the resulting scheme?

Introduction to Advanced SCA - 1

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

1. Input: set of signals $(S_i)_i$ related to a sensitive data X

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

1. Input: set of signals $(S_i)_i$ related to a sensitive data X
2. Process a function f to the S_i's

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

1. Input: set of signals $(S_i)_i$ related to a sensitive data X
2. Process a function f to the S_i's
3. For every hypothesis $\tilde{X}$ on X, estimate

$$\rho_{\tilde{X}} = |\rho(\mathrm{H}(\tilde{X}), f((S_i)_i))| \ .$$

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

1. Input: set of signals $(S_i)_i$ related to a sensitive data X
2. Process a function f to the S_i's
3. For every hypothesis $\tilde{X}$ on X, estimate

$$\rho_{\tilde{X}} = |\rho(\mathrm{H}(\tilde{X}), f((S_i)_i))| \ .$$

4. Select the hypothesis that maximizes $\rho_{\tilde{X}}$.

Introduction to Advanced SCA - 1

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA [Mes00]
- t^{th}-order Shuffling: Integrated Attacks [ClavierCoronDabbous00]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA [THM07,this paper]

Note: they all follow the same outlines.

1. Input: set of signals $(S_i)_i$ related to a sensitive data X
2. Process a function f to the S_i's
3. For every hypothesis $\tilde{X}$ on X, estimate

$$\rho_{\tilde{X}} = |\rho(\mathrm{H}(\tilde{X}), f((S_i)_i))| \ .$$

4. Select the hypothesis that maximizes $\rho_{\tilde{X}}$.

Single difference: the function f.

Introduction to Advanced SCA - 2

Introduction to Advanced SCA - 2

Goal: Investigate relation between d and t and attack efficiencies.

Introduction to Advanced SCA - 2

Goal: Investigate relation between d and t and attack efficiencies.
Need a few assumptions:

Introduction to Advanced SCA - 2

Goal: Investigate relation between d and t and attack efficiencies.
Need a few assumptions:

- The correlation coefficient ρ_X corresponding to the correct hypothesis is a sound estimator of the attack efficiency [MangardOswaldPopp06,ProuffRivainBévan09,SP06]

Introduction to Advanced SCA - 2

Goal: Investigate relation between d and t and attack efficiencies.
Need a few assumptions:

- The correlation coefficient ρ_X corresponding to the correct hypothesis is a sound estimator of the attack efficiency [MangardOswaldPopp06,ProuffRivainBévan09,SP06]
- [Hamming Weight Leakage Model] the leakage signal S_i produced by the processing of a variable D_i satisfies:

$$S_i = \delta_i + \beta_i \cdot \mathrm{H}(D_i) + N_i \quad \text{with } N_i \sim \mathcal{N}(0, \sigma) \ .$$

HO-SCA against Higher Order Masking

HO-SCA against Higher Order Masking

Context: a sensitive variable X is split into $d+1$ shares M_0, ..., M_d

HO-SCA against Higher Order Masking

Context: a sensitive variable X is split into $d+1$ shares M_0, ..., M_d

Notation: S_i is the signal related to M_i.

HO-SCA against Higher Order Masking

Context: a sensitive variable X is split into $d+1$ shares M_0, ..., M_d

Notation: S_i is the signal related to M_i.

Function f is a normalized product:

$$f(S_0, \cdots, S_d) = C_d(X) = \prod_{i=0}^{d} (S_i - \mathrm{E}[S_i]) \ .$$

HO-SCA against Higher Order Masking

Context: a sensitive variable X is split into $d+1$ shares M_0, ..., M_d

Notation: S_i is the signal related to M_i.

Function f is a normalized product:

$$f(S_0, \cdots, S_d) = C_d(X) = \prod_{i=0}^{d} (S_i - \mathrm{E}[S_i]) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_X = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

Integrated SCA Against Shuffling

Integrated SCA Against Shuffling

Context: the signal S containing information about X is randomly spread over t different signals S_1, ..., S_t.

Integrated SCA Against Shuffling

Context: the signal S containing information about X is randomly spread over t different signals S_1, ..., S_t.

Function f is an Integrated signal:

$$f(S_1, \cdots, S_t) = S_1 + S_2 + ... + S_t$$

Note: the sum always contains the term S.

Integrated SCA Against Shuffling

Context: the signal S containing information about X is randomly spread over t different signals S_1, ..., S_t.

Function f is an Integrated signal:

$$f(S_1, \cdots, S_t) = S_1 + S_2 + ... + S_t$$

Note: the sum always contains the term S.

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_X = \frac{1}{\sqrt{t}\sqrt{1 + cst_2 \cdot \sigma^2}} .$$

Advanced SCA *vs* Masking-and-Shuffling

Advanced SCA *vs* Masking-and-Shuffling

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Advanced SCA *vs* Masking-and-Shuffling

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((S_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} \mathrm{C}(S_{i_0},\cdots,S_{i_d}) \ .$$

Note: the sum always contains the term $C_d(X)$.

Advanced SCA *vs* Masking-and-Shuffling

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((S_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} C(S_{i_0},\cdots,S_{i_d}) \ .$$

Note: the sum always contains the term $C_d(X)$.

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_X = \frac{1}{\sqrt{\#I}}\rho(d,\sigma) \ .$$

Iterated Block Cipher

Iterated Block Cipher

Goal: protect block ciphers iterating round function in the form:

$$\lambda \quad \circ \quad \gamma \quad [p \oplus k] \quad] \,,$$

k: round key
p: intermediate state of the ciphering
γ: non-linear layer composed of S-boxes
λ: linear layer composed of L atomic operations.

Outlines of the Scheme

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:
  - Masking: small d' (masking only exist for $d' \leq 3$).

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:
  - Masking: small d' (masking only exist for $d' \leq 3$).
  - Shuffling: great t is required.

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:
  - Masking: small d' (masking only exist for $d' \leq 3$).
  - Shuffling: great t is required.
- Interface between λ and γ:

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:
  - Masking: small d' (masking only exist for $d' \leq 3$).
  - Shuffling: great t is required.
- Interface between λ and γ:
  - Beginning of γ : convert d-masking of data into d'-masking.

Outlines of the Scheme

Constraint: solutions to protect S-box computations with d'-masking only exist for $d' \leq 3$.

- Linear Layer λ
  - Masking: large d (masking is efficient in linear context)
  - Shuffling: small order is sufficient – deterministic function of d
- Non-linear layer γ:
  - Masking: small d' (masking only exist for $d' \leq 3$).
  - Shuffling: great t is required.
- Interface between λ and γ:
  - Beginning of γ : convert d-masking of data into d'-masking.
  - End of γ : convert d'-masking of data into d-masking.

Complexity

Complexity

Input: block cipher specifications + implem. characteristics

Complexity

Input: block cipher specifications + implem. characteristics

Three security parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

Complexity

Input: block cipher specifications + implem. characteristics

Three security parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

Complexity for one round

- Precomputations (random permutations,lookup-tables): $PreComp(t, d, d')$
- Protected Round (layers γ and λ): $RoundSec(t, d, d')$
- Protected Block Cipher:

$$PreComp(t, d, d') + RoundSec(t, d, d') \times \text{nbr of rounds}$$

Security

Security

Input: block cipher specifications + implem. characteristics

Security

Input: block cipher specifications + implem. characteristics

Three complexity parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

Security

Input: block cipher specifications + implem. characteristics

Three complexity parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

4 attack pathes have been identified.

- Targeting the t^{th}order shuffled d^{th}-masking
  - For γ: $\rho_1(t,d) = \frac{1}{\sqrt{t}}\rho(d,\sigma)$
  - For λ (split into L sub-layers): $\rho_2(t,d') = \frac{1}{\sqrt{\binom{(d+1)\cdot L}{d+1}}}\rho(d,\sigma)$
- Targeting the t^{th}order shuffled d'^{th}-masking
  - Target the d' shares simultaneously: $\rho_3(t,d') = \frac{1}{\sqrt{t}}\rho(d',\sigma)$
  - Target 2 masked data, masked with the same sum of masks: $\rho_4(t) = \frac{1}{\sqrt{t\cdot(t-1)}}\rho(2,\sigma)$.

Choice of the security parameters

Choice of the security parameters

- Fix an upper bound ρ^* [Security Bound]

Choice of the security parameters

- Fix an upper bound ρ^* [Security Bound]
- Generate the triplets (d, d', t) s.t.:

$$\max(\rho_1(t, d), \rho_2(t, d'), \rho_3(t, d'), \rho_4(t)) \leqslant \rho^* \ . \qquad (1)$$

Choice of the security parameters

- Fix an upper bound ρ^* [Security Bound]
- Generate the triplets (d, d', t) s.t.:

$$\max(\rho_1(t, d), \rho_2(t, d'), \rho_3(t, d'), \rho_4(t)) \leqslant \rho^* \ . \qquad (1)$$

- Among the 3-tuples (d, d', t) satisfying (1), chose one that minimizes

$$\mathit{PreComp}(t, d, d') + \mathit{RoundSec}(t, d, d') \times \text{nbr of rounds}$$

AES implementation on a 8051-architecture.

Table: Optimal parameters and timings according to SNR and ρ^*.

	$\mathrm{SNR} = 1$				$\mathrm{SNR} = \frac{1}{4}$			
ρ^*	t	d	d'	timings	t	d	d'	timings
10^{-1}	16	1	1	3.66×10^4	16	1	0	2.94×10^4
10^{-2}	20	2	2	6.39×10^4	16	1	1	3.66×10^4
10^{-3}	123	3	3	3.13×10^5	16	2	2	5.75×10^4
10^{-4}	12208	4	3	3.15×10^7	19	3	3	8.35×10^4

Thank you!
Questions and/or Comments?

Scheme – Linear Layer

Scheme – Linear Layer

Input: [d^{th}-masking] state $\gamma(p + k)$ masked with d new shares m'_i.

Scheme – Linear Layer

Input: [d^{th}-masking] state $\gamma(p + k)$ masked with d new shares m'_i.

Linear layer λ: [t^{th}-shuffling and d^{th}-masking]

- Signals corresponding to shares are spread over t random signals.
  - Atomic operations of λ are performed for every share

Note: no need for conversion d-masking into d'-masking.

Scheme – Linear Layer

Input: [d^{th}-masking] state $\gamma(p+k)$ masked with d new shares m'_i.

Linear layer λ: [t^{th}-shuffling and d^{th}-masking]

- Signals corresponding to shares are spread over t random signals.
  - Atomic operations of λ are performed for every share

Note: no need for conversion d-masking into d'-masking.

Output: [d^{th}-masking] state $[\lambda \circ \gamma](p+k)$ split into d shares $\lambda(m'_i)$

AES implementation on a 8051-architecture.

Table: Cycles Numbers for the different steps of the scheme for an AES implementation on a 8051-architecture.

T Generation	$\mathcal{C}_T = 112 + t\left(6 + 9\sum_{i=0}^{15}\frac{1}{t-i}\right)$
T' Generation	$\mathcal{C}_{T'} = 3q + 2^q(15 + 14q)$
Masked S-box Generation	$\mathcal{C}_{MS} = 4352d'$
Pre-computations	$\mathcal{C}_T + \mathcal{C}_{T'} + \mathcal{C}_{MS}$
γ	$\mathcal{C}_{SL} = t(55 + 37d + 18d')$
Linear Layer	$\mathcal{C}_{LL} = 676(d + 1)$
Protected Round	$\mathcal{C}_{SL} + \mathcal{C}_{LL}$
Unprotected Round	432

Complexity

Complexity

Input: block cipher specifications + implem. characteristics

Complexity

Input: block cipher specifications + implem. characteristics

Three security parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

Complexity

Input: block cipher specifications + implem. characteristics

Three security parameters:

- t: shuffling order
- d: masking order for linear layers
- d': masking order for S-box computations

Complexity for one round

Rand. Gen. [Shuffling γ]	$\mathcal{C}_T(t)$
Rand. Gen. [Shuffling λ]	$\mathcal{C}_{T'}(d)$
Masked S-box Generation	$\mathcal{C}_{MS}(d')$
Pre-computations	$\mathcal{C}_T(t) + \mathcal{C}_{T'}(d) + \mathcal{C}_{MS}(d')$
γ	$\mathcal{C}_{SL}(d, d')$
λ	$\mathcal{C}_{LL}(d)$
Protected Round	$\mathcal{C}_{SL}(d', d) + \mathcal{C}_{LL}(d)$