Blind Source Separation from Single Measurements using Singular Spectrum Analysis

CHES 2015

14.Sept.2015, Saint-Malo, France

Santos Merino del Pozo and François-Xavier Standaert

ICTEAM/ELEN/Crypto Group
Université catholique de Louvain, Belgium.

Because Noise Matters

- More noise $\rightarrow$ More side-channel measurements

Because Noise Matters

- More noise $\rightarrow$ More side-channel measurements
  - attacks become more challenging
  - critical for higher-order (HO) attacks !!

Because Noise Matters

- More noise $\rightarrow$ More side-channel measurements
  - attacks become more challenging
  - critical for higher-order (HO) attacks !!
- Ideally, *low-noise measurements*

Because Noise Matters

- More noise $\rightarrow$ More side-channel measurements
  - attacks become more challenging
  - critical for higher-order (HO) attacks !!
- Ideally, *low-noise measurements*
  - can be difficult to achieve in practice
  - architecture, countermeasures, measurement setup, ...

Because Noise Matters

- More noise $\rightarrow$ More side-channel measurements
  - attacks become more challenging
  - critical for higher-order (HO) attacks !!
- Ideally, *low-noise measurements*
  - can be difficult to achieve in practice
  - architecture, countermeasures, measurement setup, ...
- So, *preprocessing* the collected traces is always advisable

State-of-the-Art: Perks and Pitfalls

- Averaging
- Digital filtering
- PCA and LDA

State-of-the-Art: Perks and Pitfalls

- Averaging
  - ✔ easy yet effective
  - ✘ useless when exploiting HO leakages
- Digital filtering
- PCA and LDA

State-of-the-Art: Perks and Pitfalls

- Averaging
  - ✔ easy yet effective
  - ✘ useless when exploiting HO leakages
- Digital filtering
  - ✔ relevant for HO analysis
  - ✘ not trivial to design
- PCA and LDA

State-of-the-Art: Perks and Pitfalls

- Averaging
  - ✔ easy yet effective
  - ✘ useless when exploiting HO leakages
- Digital filtering
  - ✔ relevant for HO analysis
  - ✘ not trivial to design
- PCA and LDA
  - ✔ intuitive and easy to implement
  - ✘ requires profiling, extension to HO analysis?

Our Solution

- *Blind source separation using Singular Spectrum Analysis* (SSA)

Our Solution

- *Blind source separation using Singular Spectrum Analysis* (SSA)
- Disregarded in the context of side-channel analysis
- Cool features from the attackers point-of-view
  - working in a per-trace fashion
  - being readily applied to HO scenarios
  - not requiring proficiency in signal processing
  - not needing a profiling stage

Outline

- Singular Spectrum Analysis 101
- Experimental Results
  - Masked software
  - Unprotected hardware
- Conclusions

SSA 101 - Decomposition

So you got a noisy leakage trace $\boldsymbol{\ell} = \left(\ell^1, \ldots, \ell^N\right)$

- First, take $W = \lfloor \log(N)^c \rfloor$ with $c \in [1.5, 3]$,
- define $D = N - W + 1$ delayed vectors

SSA 101 - Decomposition

So you got a noisy leakage trace $\boldsymbol{\ell} = \left(\ell^1, \ldots, \ell^N\right)$

- First, take $W = \lfloor \log(N)^c \rfloor$ with $c \in [1.5, 3]$,
- define $D = N - W + 1$ delayed vectors

$$\begin{matrix} \ell^1 \\ \ell^2 \\ \vdots \\ \ell^W \end{matrix}$$

SSA 101 - Decomposition

So you got a noisy leakage trace $\boldsymbol{\ell} = \left(\ell^1, \ldots, \ell^N\right)$

- First, take $W = \lfloor \log(N)^c \rfloor$ with $c \in [1.5, 3]$,
- define $D = N - W + 1$ delayed vectors

$$\begin{matrix} \ell^1 & \ell^2 \\ \ell^2 & \ell^3 \\ \vdots & \vdots \\ \ell^W & \ell^{W+1} \end{matrix}$$

SSA 101 - Decomposition

So you got a noisy leakage trace $\boldsymbol{\ell} = \left(\ell^1, \ldots, \ell^N\right)$

- First, take $W = \lfloor \log\left(N\right)^c \rfloor$ with $c \in [1.5, 3]$,
- define $D = N - W + 1$ delayed vectors

$$\begin{matrix} \ell^1 & \ell^2 & \cdots & \ell^D \\ \ell^2 & \ell^3 & \cdots & \ell^{D+1} \\ \vdots & \vdots & \ddots & \vdots \\ \ell^W & \ell^{W+1} & \cdots & \ell^N \end{matrix}$$

SSA 101 - Decomposition

So you got a noisy leakage trace $\boldsymbol{\ell} = \left(\ell^1, \ldots, \ell^N\right)$

- First, take $W = \lfloor \log{(N)}^c \rfloor$ with $c \in [1.5, 3]$,
- define $D = N - W + 1$ delayed vectors
- and then build the so-called trajectory matrix $\mathbf{L}$

$$\mathbf{L} = \begin{pmatrix} \ell^1 & \ell^2 & \ldots & \ell^D \\ \ell^2 & \ell^3 & \ldots & \ell^{D+1} \\ \vdots & \vdots & \ddots & \vdots \\ \ell^W & \ell^{W+1} & \ldots & \ell^N \end{pmatrix}$$

SSA 101 - Decomposition

Compute the eigenvalues of $\mathbf{L}\mathbf{L}^\top$

- $(\lambda_1 \geq \cdots \geq \lambda_d)$, the so-called *singular spectrum*
- $d = W$ if none of them is zero

together with the corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$

SSA 101 - Decomposition

Compute the eigenvalues of $\mathbf{L}\mathbf{L}^\top$

- $(\lambda_1 \geq \cdots \geq \lambda_d)$, the so-called *singular spectrum*
- $d = W$ if none of them is zero

together with the corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$

The SVD decomposition of $\mathbf{L}$ is

$$\mathbf{L} = \tilde{\mathbf{L}}_1 + \cdots + \tilde{\mathbf{L}}_d,$$

such that $\tilde{\mathbf{L}}_i = \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^\top$ and $\mathbf{v}_i = \dfrac{\mathbf{L}^\top \mathbf{u}_i}{\sqrt{\lambda_i}}$

SSA 101 - Reconstruction

Now, we are ready to extract the underlying components of $\boldsymbol{\ell}$

- Each $\tilde{\mathbf{L}}_i$ matrix is transformed into the i-th component

$$\tilde{\boldsymbol{\ell}}_i = \left(\tilde{\ell}_i^1, \ldots, \tilde{\ell}_i^N\right)$$

SSA 101 - Reconstruction

Now, we are ready to extract the underlying components of $\boldsymbol{\ell}$

- Each $\tilde{\mathbf{L}}_i$ matrix is transformed into the i-th component

$$\tilde{\boldsymbol{\ell}}_i = \left(\tilde{\ell}_i^1, \ldots, \tilde{\ell}_i^N\right)$$

- Trivial when $\tilde{\mathbf{L}}_i$ is a Hankel matrix, i.e.,

$$\tilde{\mathbf{L}}_i = \begin{pmatrix} \tilde{\ell}_i^1 & \tilde{\ell}_i^2 & \tilde{\ell}_i^3 & \cdots \\ \tilde{\ell}_i^2 & \tilde{\ell}_i^3 & \cdots & \cdots \\ \tilde{\ell}_i^3 & \vdots & \ddots & \tilde{\ell}_i^{N-1} \\ \vdots & \vdots & \tilde{\ell}_i^{N-1} & \tilde{\ell}_i^N \end{pmatrix}$$

SSA 101 - Reconstruction

Now, we are ready to extract the underlying components of $\boldsymbol{\ell}$

- Each $\tilde{\mathbf{L}}_i$ matrix is transformed into the i-th component

$$\tilde{\boldsymbol{\ell}}_i = \left(\tilde{\ell}_i^1, \ldots, \tilde{\ell}_i^N\right)$$

- Trivial when $\tilde{\mathbf{L}}_i$ is a Hankel matrix, i.e.,

$$\tilde{\mathbf{L}}_i = \begin{pmatrix} \tilde{\ell}_i^1 & \tilde{\ell}_i^2 & \tilde{\ell}_i^3 & \cdots \\ \tilde{\ell}_i^2 & \tilde{\ell}_i^3 & \cdots & \cdots \\ \tilde{\ell}_i^3 & \vdots & \ddots & \tilde{\ell}_i^{N-1} \\ \vdots & \vdots & \tilde{\ell}_i^{N-1} & \tilde{\ell}_i^N \end{pmatrix}$$

- but since this is not the case, the so-called *hankelization* function must be applied on each $\tilde{\mathbf{L}}_i$

SSA 101 - Reconstruction

Lastly, the original leakage trace $\boldsymbol{\ell}$ can be reconstructed as

$$\boldsymbol{\ell} = \tilde{\boldsymbol{\ell}}_1 + \cdots + \tilde{\boldsymbol{\ell}}_d$$

SSA 101 - Reconstruction

Lastly, the original leakage trace $\boldsymbol{\ell}$ can be reconstructed as

- but we aim at a *signal vs. noise decomposition*

$$\boldsymbol{\ell} = \tilde{\boldsymbol{\ell}}_1 + \cdots + \tilde{\boldsymbol{\ell}}_d$$

SSA 101 - Reconstruction

Lastly, the original leakage trace $\boldsymbol{\ell}$ can be reconstructed as

- but we aim at a *signal vs. noise decomposition*
- $\mathcal{I} = \{1, \ldots, d\}$ is partitioned into $\mathcal{I}_{\text{signal}}$ and $\mathcal{I}_{\text{noise}}$,

$$\boldsymbol{\ell} = \tilde{\boldsymbol{\ell}}_1 + \cdots + \tilde{\boldsymbol{\ell}}_d$$

SSA 101 - Reconstruction

Lastly, the original leakage trace $\boldsymbol{\ell}$ can be reconstructed as

- but we aim at a *signal vs. noise decomposition*
- $\mathcal{I} = \{1, \ldots, d\}$ is partitioned into $\mathcal{I}_{\text{signal}}$ and $\mathcal{I}_{\text{noise}}$, so

$$\boldsymbol{\ell} = \sum_{i \in \mathcal{I}_{\text{signal}}} \tilde{\boldsymbol{\ell}}_i + \sum_{i \in \mathcal{I}_{\text{noise}}} \tilde{\boldsymbol{\ell}}_i$$

SSA 101 - Reconstruction

Lastly, the original leakage trace $\boldsymbol{\ell}$ can be reconstructed as

- but we aim at a *signal vs. noise decomposition*
- $\mathcal{I} = \{1, \ldots, d\}$ is partitioned into $\mathcal{I}_{\mathsf{signal}}$ and $\mathcal{I}_{\mathsf{noise}}$, so

$$\boldsymbol{\ell} = \sum_{i \in \mathcal{I}_{\mathsf{signal}}} \tilde{\boldsymbol{\ell}}_i + \sum_{i \in \mathcal{I}_{\mathsf{noise}}} \tilde{\boldsymbol{\ell}}_i$$

Criteria

- $\mathcal{I}_{\mathsf{noise}} \rightarrow$ small singular values producing a slowly decreasing sequence
- $\mathcal{I}_{\mathsf{signal}} \rightarrow$ the remaining ones ☺

Experimental Results

Two experimental platforms

- Atmel 8-bit μC (ATMega644p)
- Spartan-6 FPGA (SAKURA-G)

Experimental Results

Two experimental platforms

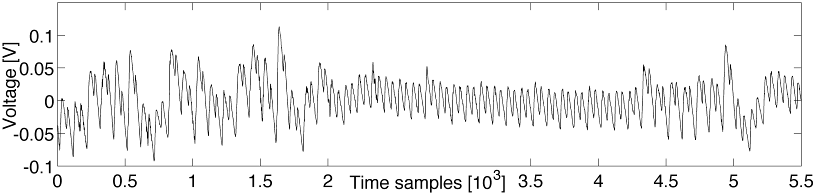
- Atmel 8-bit μC (ATMega644p)
  - First-order boolean masking scheme of AES
  - High Signal-to-Noise Ratio
  - Profiling is allowed
- Spartan-6 FPGA (SAKURA-G)

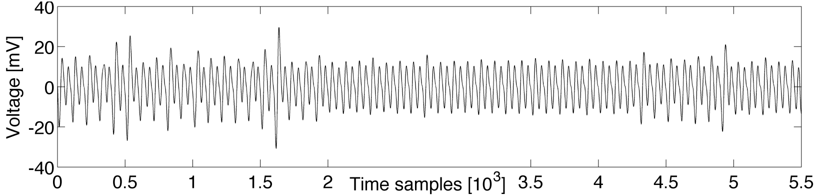
Experimental Results

Two experimental platforms

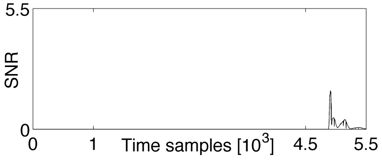
- Atmel 8-bit μC (ATMega644p)
  - First-order boolean masking scheme of AES
  - High Signal-to-Noise Ratio
  - Profiling is allowed
- Spartan-6 FPGA (SAKURA-G)
  - Unprotected implementation of PRESENT-80
  - Low Signal-to-Noise Ratio
  - Small peak-to-peak signal $\rightarrow$ quantization noise
  - Profiling is not allowed

Experimental Results - Masked software



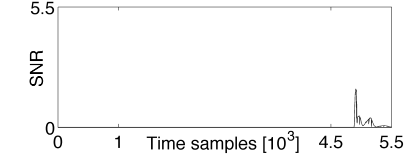


Experimental Results - Masked software

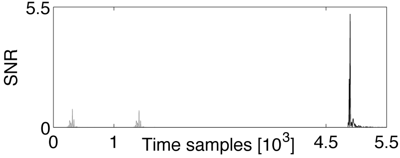


Signal-to-Noise ratio (raw)   Signal-to-Noise ratio (SSA)

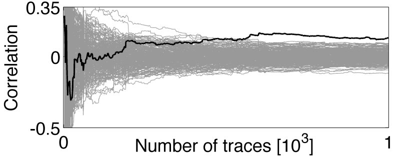
Experimental Results - Masked software



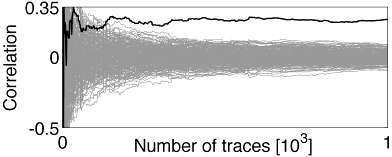
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)

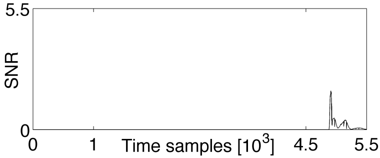


Bivariate MCP-DPA (raw)

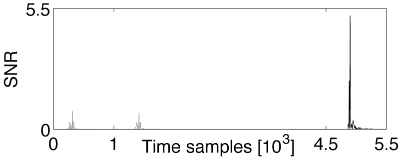


Bivariate MCP-DPA (SSA)

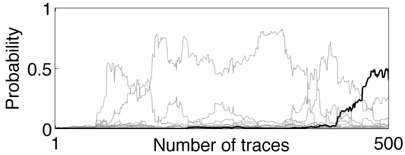
Experimental Results - Masked software



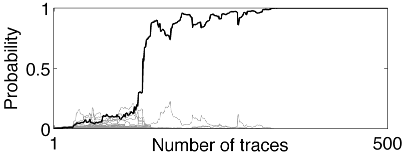
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)

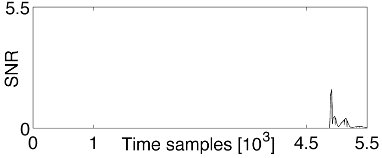


Bivariate TA (raw)

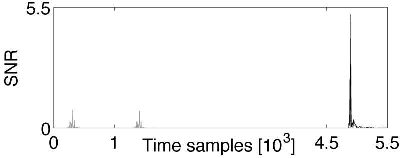


Bivariate TA (SSA)

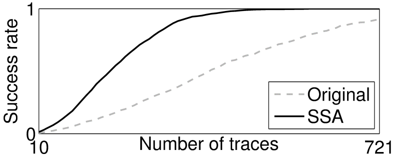
Experimental Results - Masked software



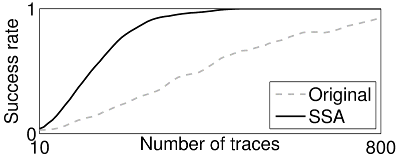
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)

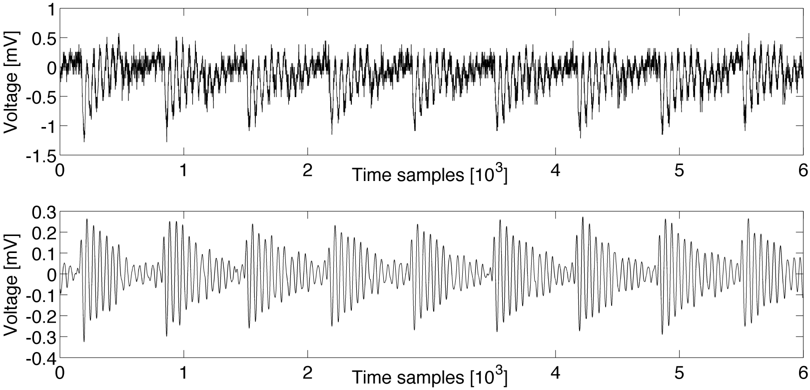


SR of bivariate MCP-DPA

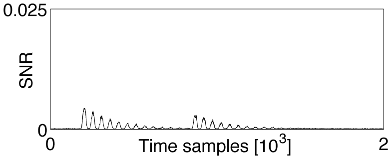


SR of bivariate TA

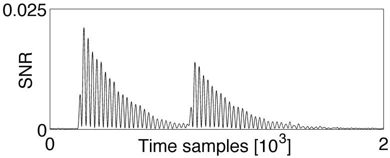
Experimental Results - Unprotected hardware



Experimental Results - Unprotected hardware

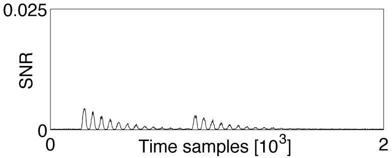


Signal-to-Noise ratio (raw)

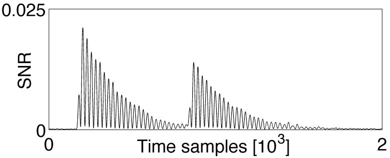


Signal-to-Noise ratio (SSA)

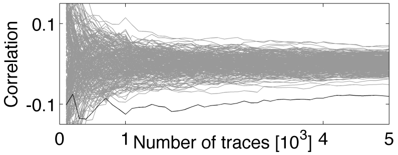
Experimental Results - Unprotected hardware



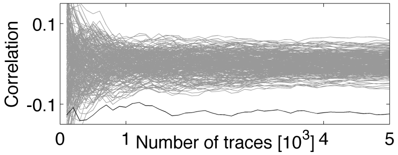
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)

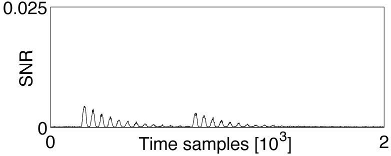


CPA using HD model (raw)

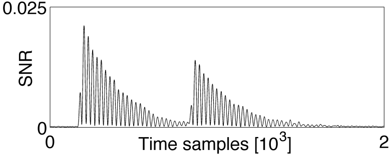


CPA using HD model (SSA)

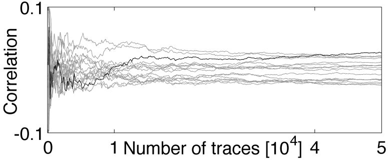
Experimental Results - Unprotected hardware



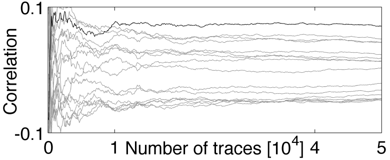
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)

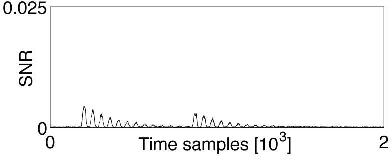


MCC-DPA (raw)

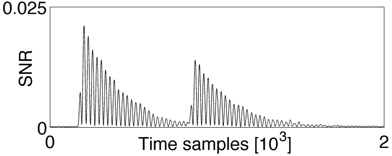


MCC-DPA (SSA)

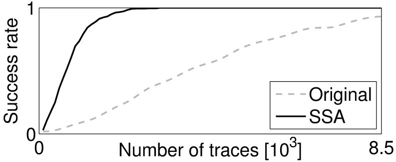
Experimental Results - Unprotected hardware



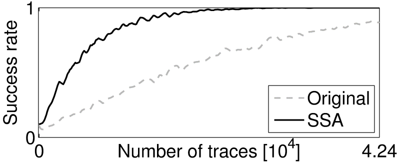
Signal-to-Noise ratio (raw)



Signal-to-Noise ratio (SSA)



SR of CPA using HD model



SR of MCC-DPA

Conclusions

- SSA in the context of side-channel analysis
  - intuitive, easy to use
    - *window length* $\rightarrow$ standard rule-of-thumb
    - *reconstruction* $\rightarrow$ visual inspection of components
  - works in a per-trace fashion
    - on-the-fly filtering
    - easily integrated into measurement frameworks
  - effective
    - SNR gains up to a factor of 4
    - attacks with reduced measurement complexity
- Future work:
  - more challenging scenarios (high noise + masking in hardware)
  - distinguish components at same frequencies?

Questions?