Support Vector Regression: Exploiting Machine Learning Techniques for Leakage Modeling

Dirmanto Jap [1] Marc Stöttinger [2] Shivam Bhasin [2]

[1]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

[2]Temasek Laboratories at Nanyang Technological University, Singapore

14 June 2015

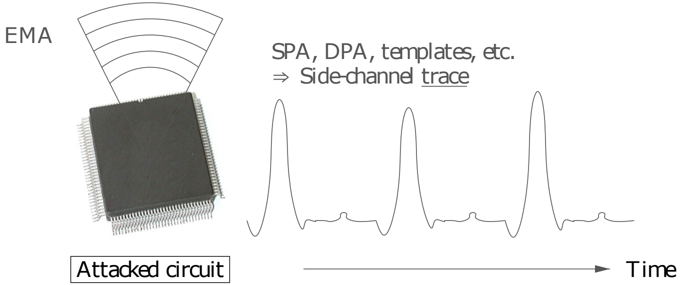
Outline

- Introduction
- Background
- Methods
- Experiments
- Conclusion

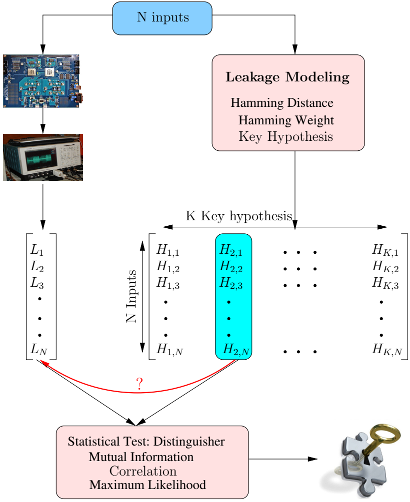
Introduction

- Side-channel analysis exploits physical leakage of the cryptographic device
- It has two main components, leakage modeling and distinguisher
- More research efforts have been focused on distinguisher
- Leakage is mainly modeled with Hamming weight, Hamming distance, bitwise, *etc*

Introduction



Introduction



Side-Channel Analysis

- Side-channel analysis can be mainly classified into profiling and non-profiling based attacks
- In non-profiling attacks, the attacker tries to exploit statistical dependency (*i.e.,* Correlation Power Analysis, Mutual Information Analysis)
- In profiling attacks, the attacker's goal is to characterize the device (*i.e.,* Template Attacks, Stochastic Approach)

Background

- The side-channel leakage can be mainly decomposed into the deterministic part and the randomized part
- Given the plaintext (x) and the key (k), the leakage for intermediate value $IV_{x,k} = f(x,k)$ is given by:

$$T_{x,k} = L(f(x,k)) + \epsilon,$$

- L is the leakage function that maps the intermediate value to its side-channel leakage $T_{x,k}$ and ϵ is the (assumed) mean free Gaussian noise ($\epsilon \sim N(0, \sigma^2)$)

Profiling Based Attacks

- These attacks are considered as the strongest attacks
- However, this is based on the assumption that the profile is built correctly
- It could be either by classification (*i.e.*, TA) or by regression (*i.e.*, SA)

Classical Profiling Attack

- Template Attacks (TA)
    - A template is constructed for each intermediate value
    - The template consists of the pair (μ, Σ)
- Stochastic Approach (SA)
    - The deterministic part of the leakage is determined using linear regression based on the subspace representation of the intermediate value
    - Different subspace are for example: F_2 which uses HW or HD, F_9 which is bitwise representation, and F_{256} which is similar to generic template model
    - Only one noise covariance matrix is used

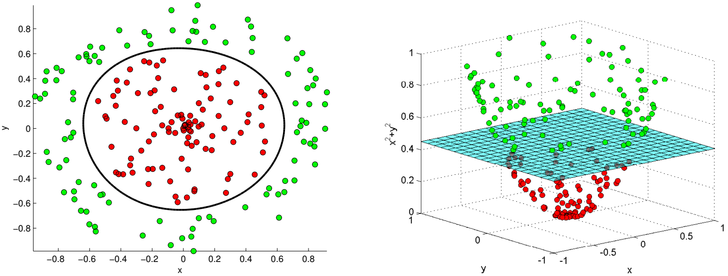
Machine Learning in Side-Channel Analysis

- Machine learning has been adopted for profiling attacks
- It is used mainly for a leakage characterization or a distinguisher
- Previous works have shown some promising results
- Commonly used learning algorithms include Support Vector Machine (SVM) and Random Forest (RF)

Support Vector Machine

- SVM have been compared with TA under different attack scenarios
- It is shown to be more robust against noise and requires less attack traces
- It is used for classification, based on separating hyperplane
- It uses soft margin to deal with non-separable data and kernel trick to deal with non-linearity issue

Support Vector Machine



(a) SVM on original data

(b) Mapping to higher dimension

Figure : How SVM performs linear classification on non-linear data, by mapping it to higher dimension space.

Support Vector Machine

- $\phi(t)$: transformation into higher dimension, might be impractical
- Primal form

$$\arg\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \text{ ,s.t: } c_i(\langle w, \boldsymbol{\phi(t_i)}\rangle + b) \geq 1 - \xi_i$$

- $K(t_i, t_j) = \langle\phi(t_i), \phi(t_j)\rangle$, can be expressed as

Kernel name	Kernel function
Linear	$K(t_i, t_j) = {t_i}^T t_j$
Radial basis function	$K(t_i, t_j) = \exp(\gamma\|t_i - t_j\|^2)$
Polynomial	$K(t_i, t_j) = (t_i \cdot t_j)^d$

- Dual form

$$\arg\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_{j,k}\alpha_j\alpha_k c_j c_k \boldsymbol{K(t_j, t_k)},$$

$$\text{s.t: } \alpha_i \leq C, \sum_i \alpha_i c_i = 0$$

Support Vector Regression

- The concept is based on support vectors like in SVM, but uses them for soft margins in the regression process instead of classification
- Additional parameter, ε, is required, to compute the loss function

Support Vector Regression

The problem in SVR is to determine $\bar{L}(\vec{a}) = \langle \vec{w}, \phi(\vec{a}) \rangle + b$, where $|\bar{L}(\vec{a}) - t| \leq \varepsilon$, which could be formulated as:

$$\arg\min_{w,b} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

subject to:

$$t_i - \langle \vec{w}, \phi(\vec{a_i}) \rangle - b \leq \varepsilon + \xi_i$$
$$\langle \vec{w}, \phi(\vec{a_i}) \rangle + b - t_i \geq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

Support Vector Regression

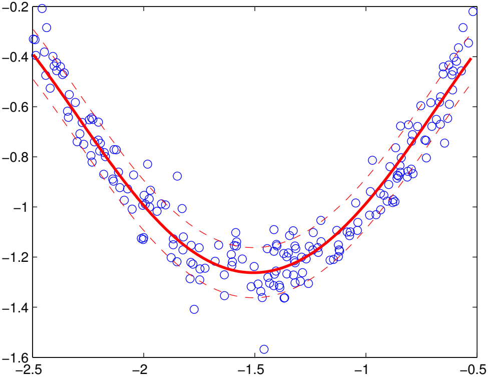


Figure : SVR on non-linear data, the dash line indicates the ε tube $(\bar{L} \pm \varepsilon)$

Support Vector Regression

- The method is done in similar manner like SA
- Replace the linear regression with SVR during the model building process to describe the deterministic part of the leakage
- To deal with parameter tuning, the heuristic method from Cherassky and Ma[1] is used

[1] V. Cherkassky and Y. Ma. Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks, 17(1):113-126, 2004

Experiments

- The experiment was done on forward AES implementation running on a standard 8-Bit μC implementation
- Exploit the power side-channel leakage from the first round Sbox output
- This is the most common target for SCA, due to its non-linear property.
- Guessing entropy is used as comparison metric

Evaluating the Quality of Leakage Modeling Using CPA

- To compare the quality of the model, Correlation Power Analysis (CPA) is used
- A set of 50000 traces from AES implementation are used
- The traces are used to estimate model using SA with F_9 (basic), denoted SA9 as well as F_{256} (maximum), denoted SA256, compared with SVR

Evaluating the Quality of Leakage Modeling Using CPA

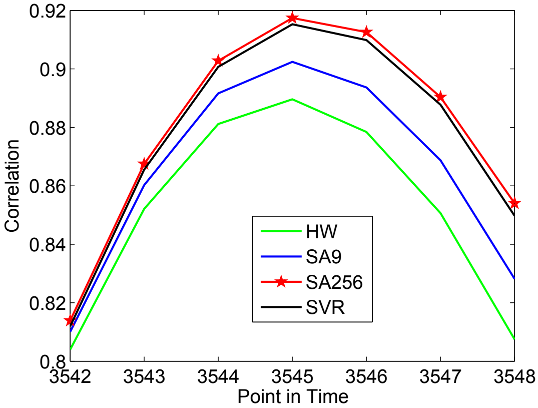
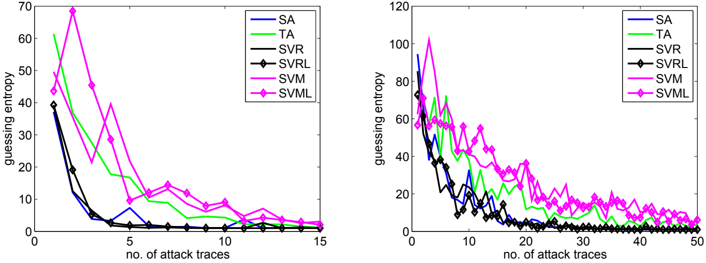


Figure : CPA of different leakage model

Evaluation of Attack on Noisy Traces

- The noise was simulated by adding white Gaussian noise to the captured power traces
- Using 50K power traces, additional sets with an artificial noise margins generated with standard deviation σ of the μC power traces: 2.5 σ (SNR 30 dB) and 8 σ (SNR 20 dB)
- Fix training set 40K and the remaining 10K was used for the evaluation of the attack

Evaluation of Attack on Noisy Traces



(a) SNR 30dB (b) SNR 20dB

Figure : Guessing entropy for different noise level

Evaluation of Attack on Different Subspaces

- Investigate inter-bit dependent leakage
- The experiment for SA is done using different subspaces (SAi uses F_i subspace)
- For SVR, only 8-bit dimensional model is used
- The experiments are done using original traces and simulated traces

Evaluation of Attack on Different Subspaces

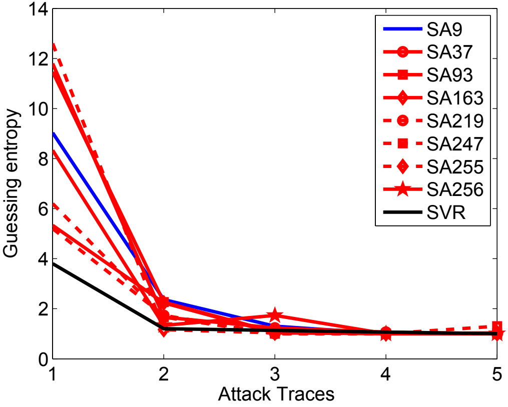
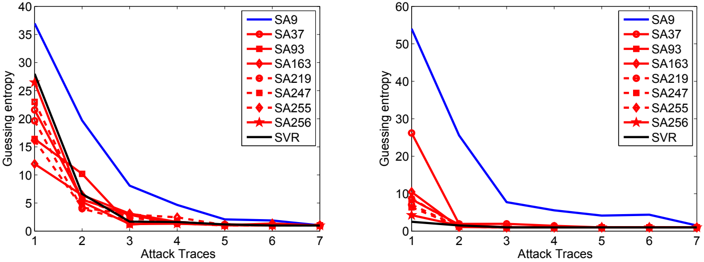


Figure : Comparison of different subspaces

Evaluation of Attack on Different Subspaces



(a) with equal additional coefficients

(b) with irregular additional coefficients

Figure : Guessing entropy on simulated data

Discussion

- The kernel trick of SVR can be used to generalize the leakage model
- When the noise level is low, SVR could perform better than SA with lower subspace, and approach the performance of SA256
- When moderate level of noise is present, the performance of SVR based profiling attacks is comparable with SA
- However, there could be a possibility of overfitting when the noise level is high

Conclusion

- We applied new machine learning based method for profiling based attacks
- The proposed method can construct good leakage model
- In the future, we will investigate the effectiveness on different platforms

Thank you!

Any questions?