Research on Side-channel Attack Based on Stochastic Model with Lower Leakage

Zixin Liu¹ and Ting Zhu²

1 Software College, East China University of Technology, Nanchang 330000, China.
2 Foreign Languages Institute, Nanchang Institute of Technology, Nanchang 330000, China.
Email: yuhaitan@126.com

Abstract. Cryptographic algorithm as an important means to ensure information security transmission has been widely concerned, but as the carrier of cryptographic algorithm, the security of encryption equipment is easy to be ignored. With the wide application of encryption equipment in modern society, the existence of side channel analysis technology is a great threat to information security. Stochastic approach (SA) is a method to deal with side-channel attack, based on many experimental studies, combined with Chebyshev filter, it can improve the accurate for attacking and signal to ratio.

1. Introduction

With the rapid development of the computer technology, embedded devices with encryption modules have become an important factor in protecting sensitive data. Although the algorithms used in these devices are mathematically secure, attackers can still recover the keys by analysing the power consumption leaked while the hardware is working [1] [2] [3]. With the explosive development of the field of deep learning in recent years, many researchers have used deep learning for side channel analysis. Side-channel analysis is a method for attacking keys by using sensitive information such as time consumption and power consumption generated by the encryption device during operation. Paul Kocher proposed simple power analysis (SPA) in 1996, which laid the foundation for side-channel attacks [4]. In 1999, Paul Kocher and others discovered a differential power analysis attack (DPA), using statistical rules for side-channel analysis [5]. In 2002, Suresh Chari proposed a template attack (TA, template attack), which greatly accelerated the time of side channel attacks [6]; in 2005, Werner Schindler et al. Proposed a stochastic approach (SA) [7], template attacks The stochastic method and the stochastic method are the most commonly used side channel analysis methods. They are modelled according to the noise model of the attacked device in accordance with the multivariate Gaussian distribution. In practical environments, such model assumptions tend to be idealized; noise-based multi-Gaussian models often bring singular matrix inversion problems[8]. If the leaked model of the attacked device is known, then a side channel attack using correlation coefficients will be a desirable method. If you can't find the correlation in the device leakage model, you can try to use a learning method to build a device leakage model and then attack.

2. Stochastic Approach

Stochastic method attacks also include training phases, but different from template attacks: instead of establishing multiple templates, the stochastic method trains a probability recognizer to predict the probability of guessing the correct key. The training vector of the key probability recognizer contains...
all possible noise vectors of the selected intermediate value. $I_t(\mu, k)$ has a recognized function for the generated noise vector. During the attack, a recognizer is used to calculate the probability of the attack power curve relative to the noise vector of each guessed key, where the key with the highest probability is the result of the attack.[9]

2.1. Estimated Energy Conversion Factor
Using $N_1$ power consumption curves, calculate the “data-dependent part of side channel leakage” at the POI position, also known as the “bit energy consumption conversion coefficient vector”. The stochastic model assumes that the energy consumption at time $t$ includes two parts, the data useful part and the white noise part:

$$I_t(\mu, k) = h_t(\mu, k) + R_t$$

Among them, $I_t(\mu, k)$ is the energy consumption of the plaintext $\mu$ and the key $k$ at the time $t$ of the power consumption curve, including the data-related energy consumption $h_t(\mu, k)$ and noise $R_t$. The stochastic model further assumes that $h_t(\mu, k)$ is a linear combination of data bit energy consumption:

$$h_t(\mu, k) = \sum_{i=1}^{Y} \beta_i g_i(\mu, k)$$

Where $g_i(\mu, k)$ is a selection function, which represents the $i$-th bit of an intermediate value generated during the selected encryption process (for example, the $i$-th bit output by the S-box). $\beta_i$ is the “bit energy consumption conversion coefficient”.

First use the actual energy consumption on the POI, and use linear regression to estimate the energy conversion coefficients at these locations to minimize the variance of the data to the fitted straight line, that is:

$$\varepsilon = \sum_{i=1}^{N_1} [I_t(\mu_i, k) - h_t(\mu_i, k)]^2$$

$$= \sum_{i=1}^{N_1} [I_t(\mu_i, k) - \sum_{j=1}^{Y} \beta_j g_j(\mu_i, k)]^2$$

$$= ||[I_t(\mu_i, k) - Ab]||^2$$

Where $A$ is a matrix with $[N_1, Y]$, $A_{ij} = g_j(\mu_i, k)$, $b = (\beta_1, \beta_2, \ldots, \beta_Y)$. Then let the gradient be $0, I_t = Ab$

$$\frac{\partial \varepsilon}{\partial b} = (I_t - Ab) = 0$$

Since matrix $A$ is not a square matrix, the solution method is:

$$b = (A^T A)^{-1} A^T I_t$$

2.2. Multiple Gaussian Noise Model
Using other $N_2$ power consumption curves and using the linear combination formula of $h_j(\mu, k)$ bit energy consumption, calculate the data dependent energy consumption on each POI and subtract the actual energy consumption to get the noise of the power consumption curve:

$$R_t = I_j(\mu, k) - h_j(\mu, k)$$

Then according to the method of template attack, $N_2$ noise vectors are used to calculate the mean vector $\rho$ and covariance matrix $\Sigma$ of the multivariate Gaussian distribution.
2.3. Attack Model
Use $N_3$ power consumption curves $(T_1, T_2 \cdots, T_{N_3})$, and set the guess key to $k'$. The same way as the template attack is converted into a noise vector $(r_1, r_2, \cdots, r_{N_3})|k')$, and their joint likelihood probability in the Gaussian model is calculated, and the minimum likelihood probability $k'$ is used as the correct key.

$$ k' = \arg\max_{k' \in k} p(r_t|k') $$

3. Lower Leakage

3.1. SNR Application
Signal-to-Noise Ratio (SNR), $SNR = Var(E(signal))/E(Var(signal))$. The signal-to-noise ratio denotes the less features included in traces, which also means the leaking of trace.[10][11]

3.2. Chebyshev Filter
Trace is different from the frequency band occupied by noise; Type I Chebyshev filters are the most common types to use among Chebyshev filters. The gain (or amplitude) response, as a function of angular frequency of the nth-order low-pass filter is equal to the absolute value of the transfer function evaluated at: The experiment performs filtering preprocessing from the frequency domain. In the actual attack, different parameters are selected according to the data and the experimental environment (filter sampling rate, etc.).

$$ |H(\tau T)|^2 = A(T^2) = \frac{1}{1 + \left(\frac{T}{T_c}\right)^{2N}} $$

Where $T_c$ is the cutoff frequency of the Butterworth low-pass filter, and $N$ is the order [12].

$$ N = \frac{\lg \left( \frac{10^{R_f/10} - 1}{10^{R_p/10} - 1} \right)}{\lg \left( \frac{T_s}{T_p} \right)} $$

4. Experiment

4.1. Experimental Data
In the DPA Contest V2 data set, each curve is 3253 sample points (In order to decrease the dimensions, only use 2500-3000), the bandwidth is 5 GHz, and the sampling rate is 5G sample/s. The FPGA device runs the AES-128 encryption algorithm in parallel at 24 MHz. The Public data set is a known data set, including 32 random keys, and each round of keys is 20000 random plaintexts. The Template data set is attack data including 32 fixed keys. Most of the experiments published establishes in Public data.

4.2. Attack Process
The power consumption curve is filtered by algorithms such as normalization, Butterworth filter and discrete Fourier transform, and the pre-processed SNR and noise mean-variance curves are compared. Attack object: Template data set (attack data containing 32 fixed keys).

Attack result:
- Table 1, Table 2 denotes enhancement of the traces, meanwhile, wave transform always change the origin shape of the curve, and we ignore its enhancement.
Table 3 shows the results, each pre-processing can reduce the guessing entropy, where Chebyshev filters denotes the best efficiency.

| Preprocessing       | 1rd  | 2rd  | 3rd  | 4rd  | 5rd  | 6rd  | 7rd  | 8rd  | 9rd  | 10rd |
|---------------------|------|------|------|------|------|------|------|------|------|------|
| None                | 107.8| 99.5 | 83.7 | 72.8 | 62.9 | 54.3 | 52.3 | 51.5 | 50.1 | 44.8 |
| Wave Transform      | 96.2 | 82.0 | 74.0 | 75.3 | 61.3 | 53.9 | 47.6 | 53.3 | 51.0 | 43.9 |
| Chebyshev filter    | 89.0 | 63.5 | 41.5 | 39.7 | 28.3 | 13.9 | 12.3 | 14.6 | 12.5 | 8.2  |

4.3. Results

Different preprocessing methods have different effects on the data set, because of different noise distributions. The preprocessing methods in this paper are used to reduce the guessing entropy of the attack, yet, the attack parameters cannot be optimized fully. In the future, the parameter optimization finding in the side channel analysis among other public data set will continue.

5. Acknowledgments

This work is supported in part by Optimization of GIS network architecture based on cloud computing 2.0, JETRCN GDSS201904; Application of machine learning method in image quality assessment, GJJ160589.

6. References

[1] François Koeune, François-Xavier Standaert. A Tutorial on Physical Security and Side-Channel Attacks[J]. 2005.
[2] Lv J, Han Y. Enhanced DES Implementation Secure Against High-Order Differential Power Analysis in Smartcards[C]// Australasian Conference on Information Security & Privacy. 2005.
[3] CHARI, S., RAO, J. R., and ROHATGI, P., Template attacks[M]. Cryptographic Hardware and Embedded Systems–CHES 2002. ed: Springer, 2003.
[4] Potlapally N R, Raghunathan A, Ravi S, et al. Aiding Side-Channel Attacks on Cryptographic Software With Satisfiability-Based Analysis[J]. IEEE Transactions on Very Large Scale Integration Systems, 2007, 15(4):465-470.
[5] Clavier C, Danger J L, Duc G, et al. Practical improvements of side-channel attacks on AES: feedback from the 2nd DPA contest[J]. Journal of Cryptographic Engineering, 2014, 4(4):259-274.
[6] Hannan M A, Arebey M, Begum R A, et al. An automated solid waste bin level detection system using Gabor wavelet filters and multi-layer perception[J]. Resources Conservation & Recycling, 2013, Volume 72(2):Pages 33–42.
[7] Monmasson E, Cirstea M N. FPGA Design Methodology for Industrial Control Systems—A Review[J]. IEEE Transactions on Industrial Electronics, 2007, 54(4):1824-1842.
[8] Yang L J, Zhang B H, Xu-Zhen Y E. Fast Fourier transform and its applications[J]. Opto-electronic Engineering, 2004, 31:303-350.

[9] Harris F J. On the use of windows for harmonic analysis with the discrete Fourier transform[J]. IEEE Proceedings, 1978, 66(1):51-83.

[10] Griffin D, Lim J S. Signal estimation from modified short-time Fourier transform[J]. IEEE Transactions on Acoustics Speech & Signal Processing, 1984, 32(2):236-243.

[11] Yang L J, Zhang B H, Xu-Zhen Y E. Fast Fourier transform and its applications[J]. Opto-electronic Engineering, 2004, 31:303-350.

[12] Maghrebi H, PortigliattiT, ProuffE. Breaking Cryptographic Implementations Using Deep Learning Techniques[J]. 2016.