Behind the Scene of Side Channel Attacks

ASIACRYPT 2013

Victor LOMNE, Emmanuel PROUFF and Thomas ROCHE

ANSSI (French Network and Information Security Agency)

Thursday, December 3rd, 2013

Agenda

Agenda

Context

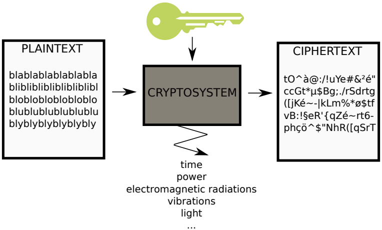
- Since the 90's, increasing use of Embedded Systems
  - 7G smartcards sold in 2012 (SIM, banking, pay-TV, ID, ...)


- Embedded Systems integrating Cryptography are susceptible to Side Channel Cryptanalysis

Side Channel Cryptanalysis [Kocher et al - Crypto99]

- A CMOS device leaks info. about its state during a computation through side-channels
  e.g.: time, power consumption, EM radiations, ...
- SCA exploit these physical leakages to guess a secret

gray-box model
(spy the computation)



Generic SCA Flow

1. Collect N side channel traces w. known inputs
   $t_1 \to Enc(p_1, k), \ldots, t_N \to Enc(p_N, k)$
2. Choose sensitive variable depend. on input & secret
   e.g. AES Sbox output $\to v_i^{\hat{k}} = S(p_i \oplus \hat{k})$
3. Choose a Leakage Model
   e.g. Hamming Weight (H)
4. Compute predictions for each key hypothesis
   $\hat{k} = 0 \quad \to H(v_1^{\hat{k}=0}), \ldots, H(v_N^{\hat{k}=0})$
   ...
   $\hat{k} = 255 \to H(v_1^{\hat{k}=255}), \ldots, H(v_N^{\hat{k}=255})$
5. Use a distinguisher to discriminate the correct key by comparing the N traces and the predictions

SCA flow and Leakage Model: 3 cases

1. Select a priori a Leakage Model
   - Hamming Weight, Hamming Distance
   - Used in classical SCA (DPA, CPA, MIA, ...)
2. Select a priori a space of Leakage Models
   - Attack will *guess* the correct model in selected space
   - Used in Linear Regression Attack (LRA)
3. Infer a Leakage Model through profiling before attack
   - A preliminary step is performed on an open copy of the device to build a leakage model for each key value
   - Used in Template Attack (TA)

Agenda

Microelectronics & Side Channel Cryptanalysis

- Moore's law:
  - Nb. of transistors on ICs doubles approx. every two years
  - CMOS processes decrease
    1995 $\rightarrow$ CMOS process 350nm / 2013 $\rightarrow$ CMOS process 22nm
- Consequence:
  - intra-chip variability increases
    $\Rightarrow$ bits leak differently ones from others
  - inter-chip variability increases
    $\Rightarrow$ two identical ICs behave differently

$\Rightarrow$ New challenges for Side Channel Attacks ?!

Our Contributions

- Propose a study on the practicality of:
  - Linear Regression Attack (LRA)
  - Template Attack (TA)
- Perform experiments on 3 different microcontrollers:
  - Device A - CMOS process 350nm - AES128 enc.
    51600 points per trace - highest SNR[1]: 0.3
  - Device B - CMOS process 130nm - AES128 enc.
    16800 points per trace - highest SNR[1]: 0.6
  - Device C - CMOS process 90nm - AES128 enc.
    12800 points per trace - highest SNR[1]: 0.09
- Use of 3 copies of each device for cross-tests

[1] SNR: Signal-to-Noise Ratio

Agenda

Linear Regression Attack [Doget et al - Cosade11]

- Leakage function $\mathfrak{L}$: models the handling of sens. var. v
  handling of $v \rightarrow \mathfrak{L}(v) + \mathfrak{B}$, with $\mathfrak{B}$ a gaussian noise

- In LRA, $\mathfrak{L}$ assumed unknown and viewed as a multivariate polynomial in the bit-coordinates v_i of v w. coefs. in $\mathbb{R}$
  $$\mathfrak{L}(v) = \underbrace{\epsilon_0 v_0 + \epsilon_1 v_1 + \ldots}_{\text{linear part}} + \underbrace{\epsilon_{0,1} v_0 v_1 + \epsilon_{0,2} v_0 v_2 + \ldots}_{\text{quadratic part}} + \underbrace{\ldots}_{\text{etc}}$$

- In LRA, guessing $\mathfrak{L}$ is hence equivalent to solve a polynomial interpolation in a noisy context
  $\Rightarrow$ use of linear regression techniques

LRA Issues

- Previous works reporting experiments on LRA consider side channel traces composed of one unique point
- In practice, side channel traces are never composed of one unique point, but rather several thousands
- Classical strategy consists in applying SCA on each time sample and to keep the key candidate maximizing the score over all time samples
- In our experiments, such a strategy did not work for LRA

Our Solution

- From our experiments, we observed that correct key k is ranked first at time samples where:
  - the distance $score(k) - \mathbb{E}[score(k)]$ is large
  - $Var[score(k)]$ is small

- We hence deduced a normalization step:
  - center the scores
  - divide by their variance
  - $normalized\_score(k) = \frac{score(k) - \mathbb{E}[score(k)]}{Var[score(k)]}$

Agenda

LRA vs. Normalized LRA (device A - 350nm)

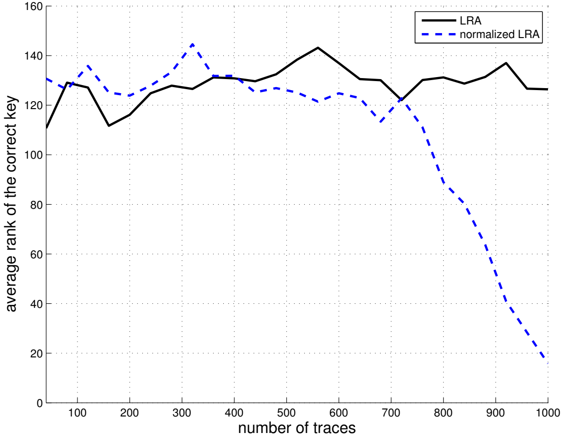


Figure: Correct key rank evolution vs. nb. of traces

LRA vs. Normalized LRA (device B - 130nm)

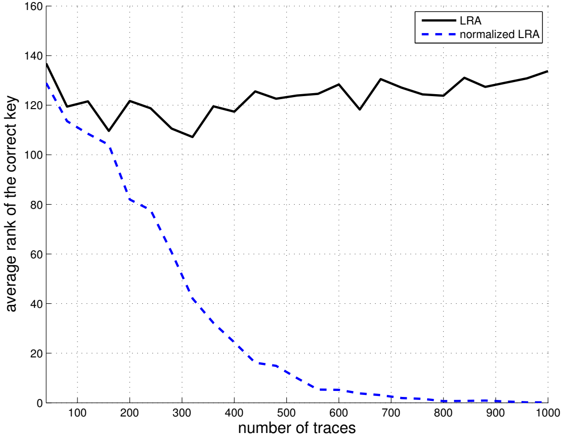


Figure: Correct key rank evolution vs. nb. of traces

LRA vs. Normalized LRA (device C - 90nm)

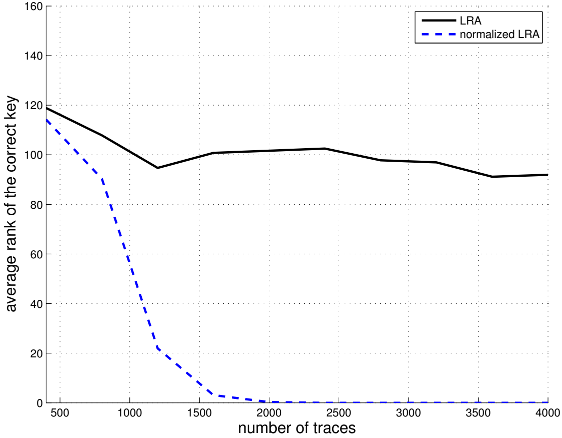


Figure: Correct key rank evolution vs. nb. of traces

Normalized LRA vs. CPA (device C - 90nm)

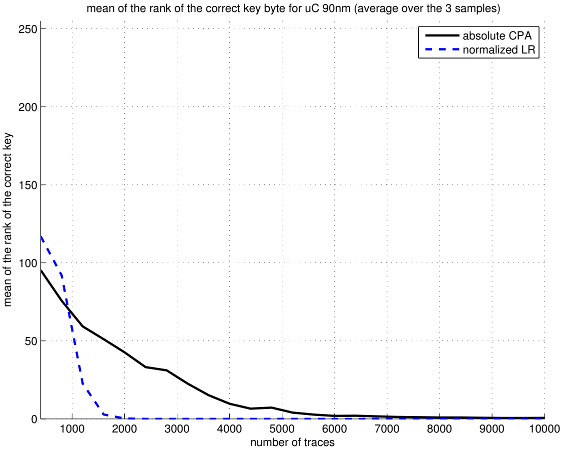


Figure: Correct key rank evolution vs. nb. of traces

Agenda

Template Attack (TA) [Chari et al - CHES02]

1. Profiling Phase (performed on an open device copy)
    1.1 Collect M side channel traces w. known inputs & keys
    $t_1 \rightarrow Enc(p_1, k_1), \ldots, t_N \rightarrow Enc(p_M, k_M)$
    1.2 Choose sensitive variable depend. on input & secret
    e.g. AES Sbox output $\rightarrow v_i^{\hat{k}} = S(p_i \oplus \hat{k})$
    1.3 Compute the pdf of the leakage for each key value
    $pdf_{\hat{k}=0}, \ldots, pdf_{\hat{k}=255}$
2. Attack Phase (performed on a device copy set at an unknown secret)
    2.1 Collect N side channel traces w. diff. inputs
    $t_1 \rightarrow Enc(p_1, k), \ldots, t_N \rightarrow Enc(p_N, k)$
    2.2 Use a maximum likelihood test to discriminate the correct key by comparing the N traces and the pdfs

Agenda

TA on device A (350nm) - copy 1 $\rightarrow$ copy 1

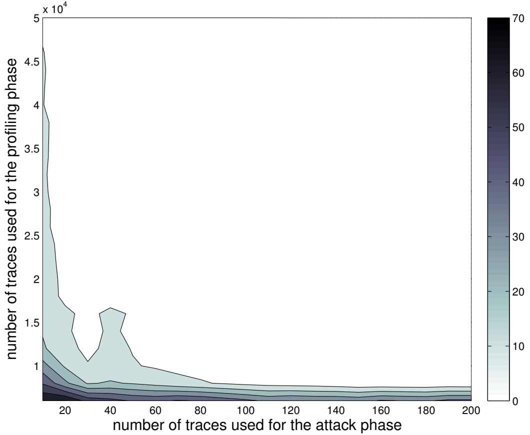


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

TA on device A (350nm) - copy 1 → copy 2

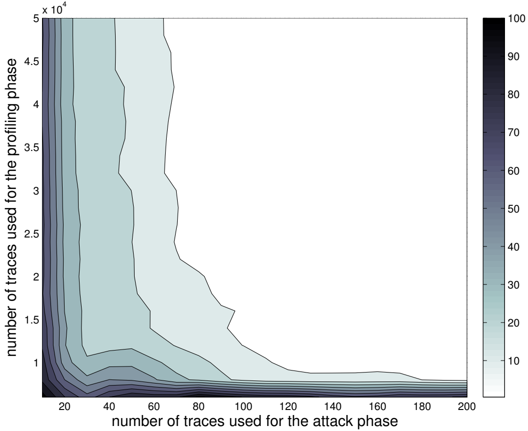


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

TA on device B (130nm) - copy 1 → copy 1

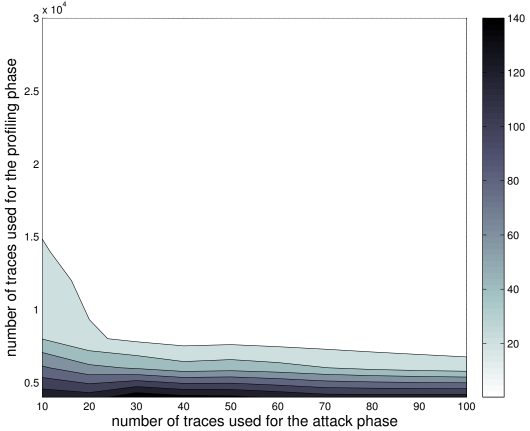


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

TA on device B (130nm) - copy 1 → copy 2

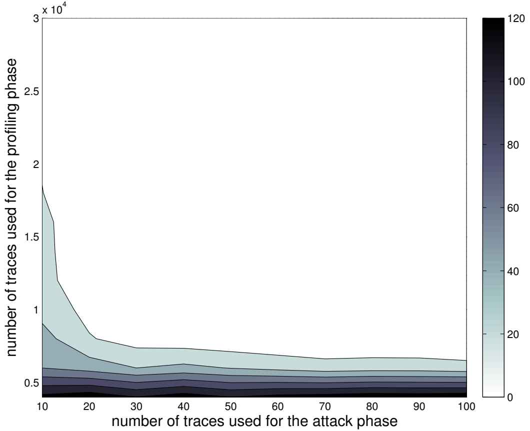


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

TA on device C (90nm) - copy 1 $\rightarrow$ copy 1

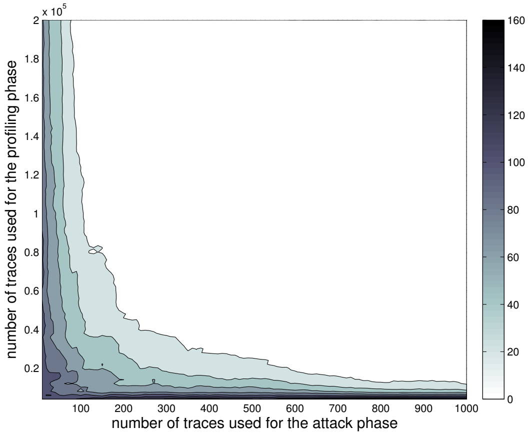


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

TA on device C (90nm) - copy 1 → copy 3

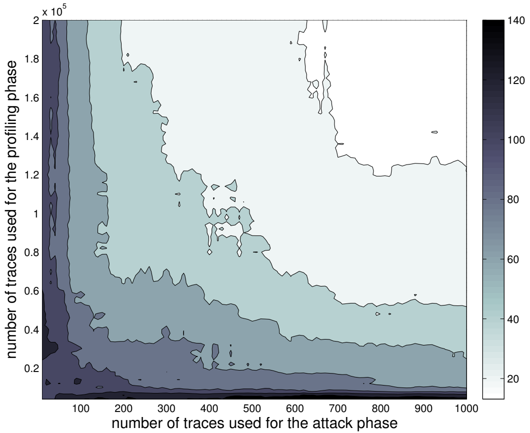


Figure: Correct key rank evolution vs. nb. of traces for the profiling phase (y-axis) and the attack phase (x-axis)

Agenda

Our Results

1. Improvement to apply LRA in a practical setup

2. Experiments show that LRA is more effective than classical SCA as CMOS process tends to nanometer scale

3. Experiments show that TA work well:
    - even if both phases are performed on diff. device copies
    - TA effectiveness outperforms unprofiled SCA

4. Partition method allowing to implement efficiently all SCA
    - algo. complexity does not depend from nb. of traces (not described in this presentation)

Some Numbers

uC	CPA	LRA	TA[2]	TA[3]
350nm	250	1000	80	10
130nm	350	800	100	10
90nm	7500	2000	700	100

Table: average nb. of traces to retrieve the correct key

[2] Template Attack inter-chip
[3] Template Attack intra-chip

Talk Finished !

Thanks for your attention !

Questions ?

full version: http://eprint.iacr.org/2013/794