Template Attack vs. Bayes Classifier

Stjepan Picek[1] Annelie Heuser[2]
Sylvain Guilley[2]

[1]KU Leuven, Belgium

[2]TELECOM-ParisTech, Paris, France

PROOFS,
Santa Barbara
August 20, 2016

Outline

Outline

Short Intro to Implementation Attacks and SCA

Implementation attacks

Implementation attacks do not aim at the weaknesses of the algorithm itself, but on the actual implementations on cryptographic devices.

- Implementation attacks can be categorized on **active** and **passive** attacks.
- In passive attacks, the device operates within its specification and the attacker just reads hidden signals.
- **Side-channel attacks** (SCA) belong into passive, non-invasive attacks.
- Side-channel attacks represent one of the most powerful category of attacks on cryptographic devices.

Profiled Attacks

- Profiled attacks have a prominent place as the most powerful among side channel attacks.
- Within profiling phase the adversary estimates leakage models for targeted intermediate computations, which are then exploited to extract secret information in the actual attack phase.
- Template Attack (TA) is the most powerful attack from the information theoretic point of view.
- TA efficiency can only be guaranteed when the template estimates are provided with an reasonable amount of traces in the profiling phase.
- Some machine learning (ML) techniques also belong to the profiled attacks.

Motivation

- When working with ML, methods used up to now belong to more powerful ML techniques.
- However, when using such powerful methods, space and time complexity grows significantly.
- Tuning phase is a long process where one cannot be sure in the results.
- It is difficult to explain on an intuitive level what is happening.
- Finally, it becomes very difficult to follow some more theoretical framework.
- Accordingly, our goal is to explore some simpler ML techniques where there is also a clear connection between those methods and TA.

Outline

Introduction to ML

- Machine learning (ML) is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory.
- Algorithms extract information from data, however, they also learn a model to discover something about the data in the future.
- Today, there exists a plenitude of ML algorithms when could choose from.

Machine Learning

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured with P, improves with experience E.

Types of ML on a Basis of Feedback

- Supervised learning - available data also include information how to correctly classify at least a part of data.
- Unsupervised learning - input data does not tell the algorithm what the clusters should be.
- Reinforcement learning.
- Active learning.

What can we do with ML

- Regression.
- Feature selection.
- Prototyping.
- Classification.
- Clustering

What can we do with ML

- Regression.
- Feature selection.
- Prototyping.
- Classification.
- Clustering

What can we do with ML

- Regression.
- Feature selection.
- Prototyping.
- Classification.
- Clustering

What can we do with ML

- Regression.
- Feature selection.
- Prototyping.
- Classification.
- Clustering

What can we do with ML

- Regression.
- Feature selection.
- Prototyping.
- Classification.
- Clustering

No Free Lunch

No Free Lunch

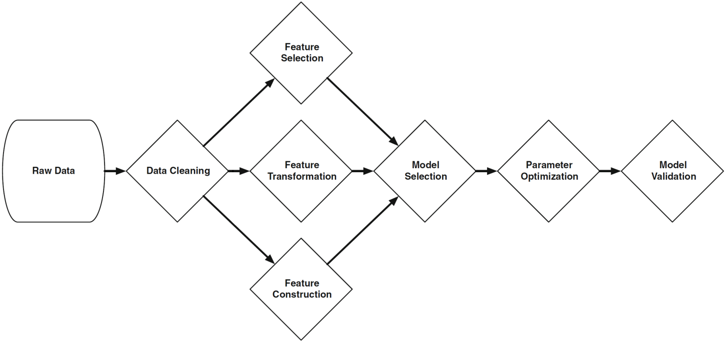
There exists no single model that works best for every problem.

- To find the best model for a certain problem, numerous algorithms and parameter combinations should be tested.
- Not even then we can be sure that we found the best model, but at least we should be able to estimate the possible trade-offs between the speed, accuracy, and complexity of the obtained models.

ML model

- Training set consists of pairs (x, y) called training examples.
- x is a feature vector, y is a label (classification value for x).
- Objective is to find function $y = f(x)$.
- if y is a real number $\rightarrow$ regression.
- y is a Boolean variable $\rightarrow$ binary classification.
- y is member of a finite set $\rightarrow$ multiclass classification.

ML architecture



Outline

Profiled Attacks

- We are particularly interested in multivariate leakage $\vec{X} = X_1, \ldots, X_D$, where D is the data dimensionality (i.e., the number of time samples per measurement trace).
- In order to guess the secret key an attacker chooses a model Y depending on a key guess k and on some known text T.
- Considering a powerful attacker, a set of N profiling traces $\vec{X}_1, \ldots, \vec{X}_N$ is used in order to estimate the leakage model beforehand, which can then be used in the attacking phase with $\vec{X}_1, \ldots, \vec{X}_Q$ traces.

Template Attack

- Given $\vec{X}_1, \ldots, \vec{X}_N$ measurements in the profiling phase the template attack (TA) consists in estimating

$$\hat{P}(\vec{X}|Y = y)$$

for all possible values of y.

- In the attack phase the attacker uses a new set of measurements $\vec{X}_1, \ldots, \vec{X}_Q$ and decides for a key $\hat{k}$ given by

$$\hat{k} = \arg\max_{k \in \mathcal{K}} \prod_{\vec{X}_1, \ldots, \vec{X}_Q} \hat{P}(\vec{X}|Y(k)).$$

Naive Bayes

- Naive Bayes is a method based on the Bayesian rule, but it works under a simplifying assumption that the predictor attributes (measurements) are mutually independent among the D features given the target class.
- Existence of highly-correlated attributes in a dataset can thus influence the learning process and reduce the number of successful predictions.

$$p(Y = y|X = x) = p(Y = y)\prod_{i=1}^{D} p(X_i = x_i|Y = y).$$

A0DE

- If the assumption of independence is violated, Naive Bayes may result in high precision loss.
- In Averaged One-Dependence Estimators there is a Super-Parent One-Dependence Estimate that relaxes the assumption of independence by making all other attributes independent given the class and one privileged attribute called the super-parent x_α.
- Since this is a weaker assumption, the bias of this model should be lower, while the variance should be higher since it is derived from higher-order probability estimates.

$$p(Y = y|X = x) = p(Y = y, x_\alpha) \prod_{i=1}^{D} p(X_i = x_i | Y_i = y_i, x_\alpha).$$

AnDE

- AnDE algorithm works by learning an ensemble of n-dependence classifiers where the prediction is obtained by aggregating the predictions of all classifiers.
- n-dependence estimator means that the probability of an attribute is conditioned by the class variable and at most n other attributes.
- In AnDE algorithm, an n-dependence classifier is constructed for every combination of n attributes where those n attributes are set as parents to all other attributes.

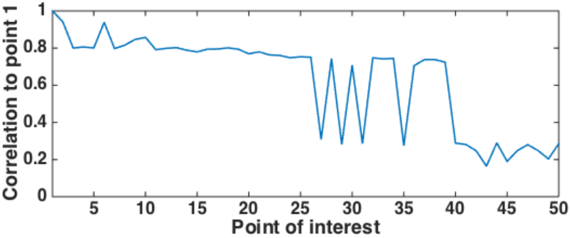
$$p(Y=y|X=x)=\sum_{s\in S^n} p(Y=y,x_s)\prod_{i=1}^{D} p(X_i=x_i|Y_i=y_i,x_s)/\binom{D}{n},$$

Outline

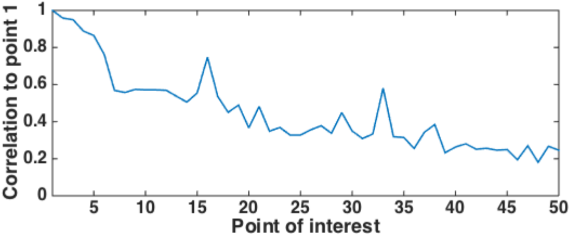
Datasets

- Datasets with 5 000, 10 000, 20 000, 30 000, 50 000, and 100 000 measurements which are randomly selected from the whole data sets.
- 2/3 of the data is for training and 1/3 for testing.
- The number of features equals 50 and the model consists either of 256 uniformly distributed classes (S-box output) or 9 binomial distributed classes (HW of the S-box output).
- DPAcontest v2 $\rightarrow$ provides measurements of an AES hardware implementation.
- DPAcontest v4 $\rightarrow$ provides measurements of a masked AES software implementation.

DPAcontest v2



DPAcontest v4



A1DE Tuning

Table: Parameter tuning

freq/m	DPAcontest	0.1	0.2	0.5	0.8	1	2	3	4	5
					9 classes					
1	v4	83.22	83.33	83.35	83.34	83.36	83.39	83.3	83.3	83.29
1	v2	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86
					256 classes					
1	v4	22.68	22.67	22.76	22.77	22.67	22.22	22.02	21.85	21.78
1	v2	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54

The frequency limit *freq* parameter denotes that all features with a frequency in the train set below this value are not used as parents, weight parameter *m* sets the base probabilities with *m*-estimation.

Verification of Results for 9 classes

Table: Testing results for 9 classes (ACC/F-Measure/AUC)

Size	Naive Bayes	*A1DE*	TA	TA (pooled)
DPAcontest v4				
5 000	65.52/65.5/91.3	78.12/78.1/96.3	19.49	62.07
10 000	67.01/67.1/91.5	81.26/81.3/97.2	52.14	76.54
20 000	68.25/66.7/91.3	83.39/83.4/97.7	75.43	77.78
30 000	67.66/67.7/91.7	84.25/84.3/97.9	77.45	78.09
50 000	67.19/67.2/91.5	84.93/84.9/98	78.71	77.85
100 000	67.29/67.3/91.7	85.55/85.6/98.1	79.91	77.83
DPAcontest v2				
5 000	10.06/10.5/50.1	25.76/10.6/50	1.29	10.07
10 000	10.94/9.9/50.1	26.06/10.8/50	1.73	8.74
20 000	7.88/9.2/50.5	27.1/11.6/50	15.48	7.64
30 000	8.81/10.4/50.3	25.6/15.5/51.7	17.66	6.66
50 000	10.21/11.6/50.4	24.3/15.8/51.2	15.99	5.88
100 000	12.44/14.1/50.6	23.79/16.3/50.5	13.20	5.98

Verification of Results for 256 classes

Table: Testing results for 256 classes (ACC/F-Measure/AUC)

Size	Naive Bayes	*A1DE*	TA	TA (pooled)
			DPAcontest v4	
5 000	15.29/14.7/91.6	10.29/8/93.7	0.23	14.89
10 000	18.26/17.1/93.4	15.65/13.7/95.5	0.32	19.68
20 000	20.21/18.3/94.5	22.56/21.2/96.9	0.52	23.65
30 000	20.88/19/94.7	28.19/27.4/97.7	9.44	25.53
50 000	21.22/19.1/95	32.06/31.5/98.2	15.63	27.47
100 000	12.44/14.1/50.6	23.71/16.8/51	21.66	29.14
			DPAcontest v2	
5 000	0.59/0.1/51	0.06/0/50	0.53	0.11
10 000	0.56/0.2/51.3	0.38/0/50	0.52	0.32
20 000	0.6/0.1/51.2	0.34/0/50	0.55	0.32
30 000	0.63/0.1/50.8	0.29/0/50	0.30	0.40
50 000	0.51/0.1/51.1	0.41/0/50	0.36	0.50
100 000	0.54/0.1/50.9	0.39/0/50	0.46	0.45

Space and Time Complexities

	Training		Testing	
	Space comp.	Time comp.	Space comp.	Time comp.
NB	$O\left(kav\right)$	$O\left(ta\right)$	$O\left(kav\right)$	$O\left(ka\right)$
A1DE	$O\left(k\binom{a}{n+1}v^{n+1}\right)$	$O\left(t\binom{a}{n+1}\right)$	$O\left(k\binom{a}{n+1}v^{n+1}\right)$	$O\left(ka\binom{a}{n}\right)$
TA	$O\left(ka^2v\right)$	$O\left(ta^2\right)$	$O\left(ka^2v\right)$	$O\left(ka^2\right)$

k is the number of classes

a is the number of features

v is the average number of values for an attribute

t is the number of training examples

n is the number of parent nodes.

Outline

Observations

- Pooled TA has a higher accuracy than TA when the profiling set is rather small.
- With the increase of the profiling set, TA becomes better than the pooled TA.
- NB is worse than pooled TA and TA when the number of measurements is high.
- A1DE is better than TA when working with DPAcontest v4.

Table: Testing results for 9 classes with an equal number of measurements.

Dataset	v4	v2
Naive Bayes	73.76	14.75
A1DE	80.67	11.76
TA	63.61	12.53
TA (pooled)	77.82	13.00

Outline

Conclusions

- Naive Bayes and A1DE give competitive results when compared with TA.
- In general, A1DE is better than Naive Bayes.
- The results seem to be particularly good when the number of measurements is low.
- Furthermore, both space and time complexity work in favor of Naive Bayes (and somewhat less A1DE).
- In our opinion, both NB and A1DE represent a viable choice and a must for the initial assessment of the ML performance.
- Since those methods are simple, also PAC learning is possible!

Questions?

Thanks for your attention!

Q?