ML Based SCA on RNS ECC Implementations using Hybrid Feature Engineering

Louiza Papachristodoulou

Joint work with

N. Mukhtar, A. Fournaris,

L. Batina, Y. Kong

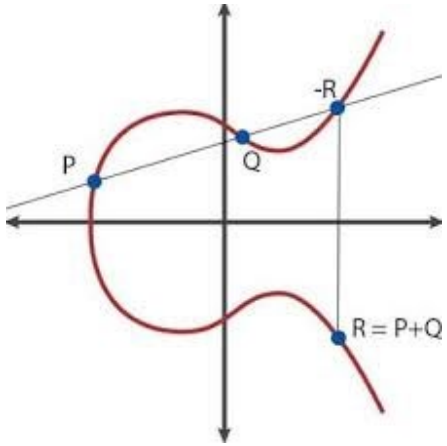
Outline

- Residue Number System as SCA countermeasure
- ML Evaluation Methodology for RNS ECC
- Hybrid Feature Engineering Approaches
- Practical Results

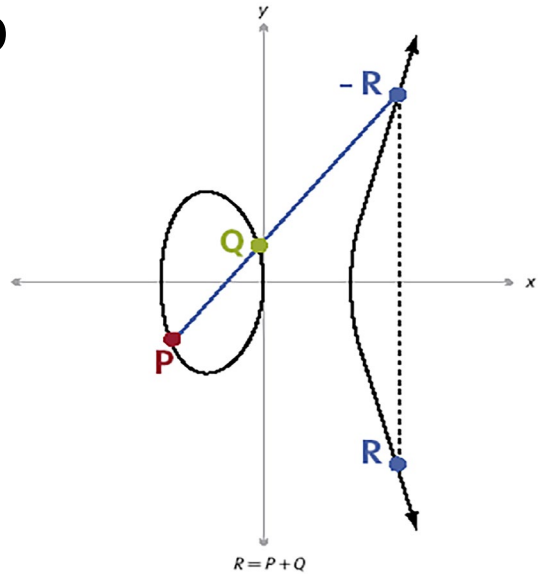
Residue Number System (RNS)

- Initially used for digital signal processing and for parallelizing computations
- A given data range can be decomposed into parallel paths of smaller dynamic ranges -> reduce complexity, computational speed and power consumption
- Adopted for RSA and finite field operations over elliptic curves
- RNS Montgomery multiplication by Bajard in 1997
- Lately (2018) used for lattice-based crypto and homomorphic encryption

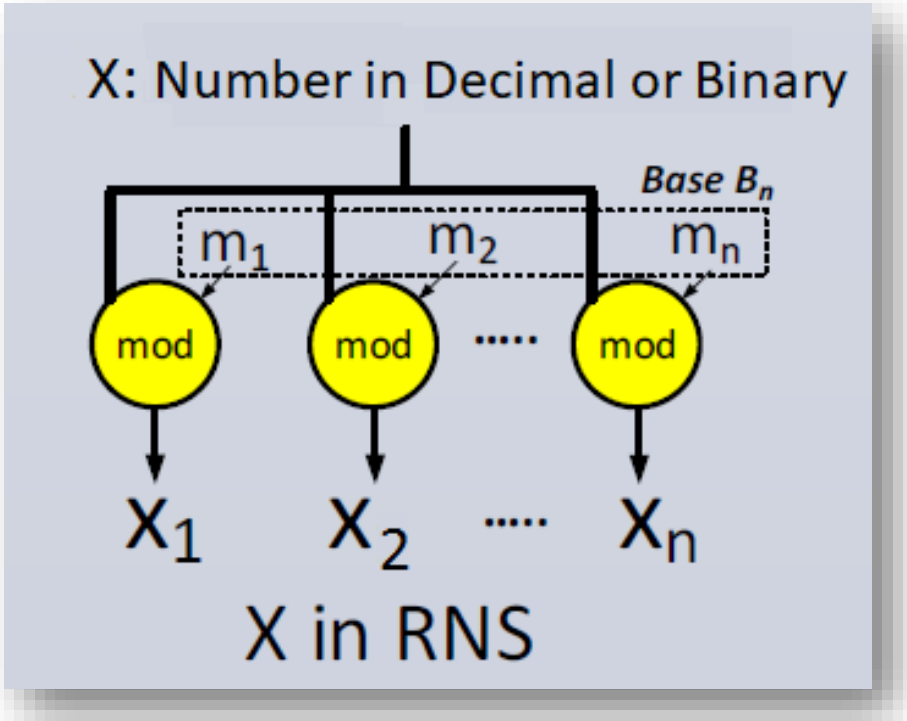
RNS & Elliptic Curve Cryptography



- Elliptic curves defined over prime fields GF(p)
- Modular operations turn easily to RNS modular operations over GF(p)
- RNS mod multiplication usually realized through RNS Montgomery multiplication to avoid modular inversion, but includes base extension
- EC scalar multiplication is the critical operation Q = kP



Residue Number System Example



$X = 50$

$(m1, m2, m3) = (3, 7, 11)$

$(x1, x2, x3) = (2, 1, 6)$

Leak Resistant Arithmetic (LRA)

- 2004: Bajard, Imbert, Liardet, Teglia
- Protection at arithmetic level, i.e. in the way we represent the numbers for internal computations
- Random choice of initial bases
- Random change of bases before and during the exponentiation -> very expensive

LRA Montgomery Power Ladder

Choose base B_n , B'_n Transform V, R to RNS format using permutat. γ_t

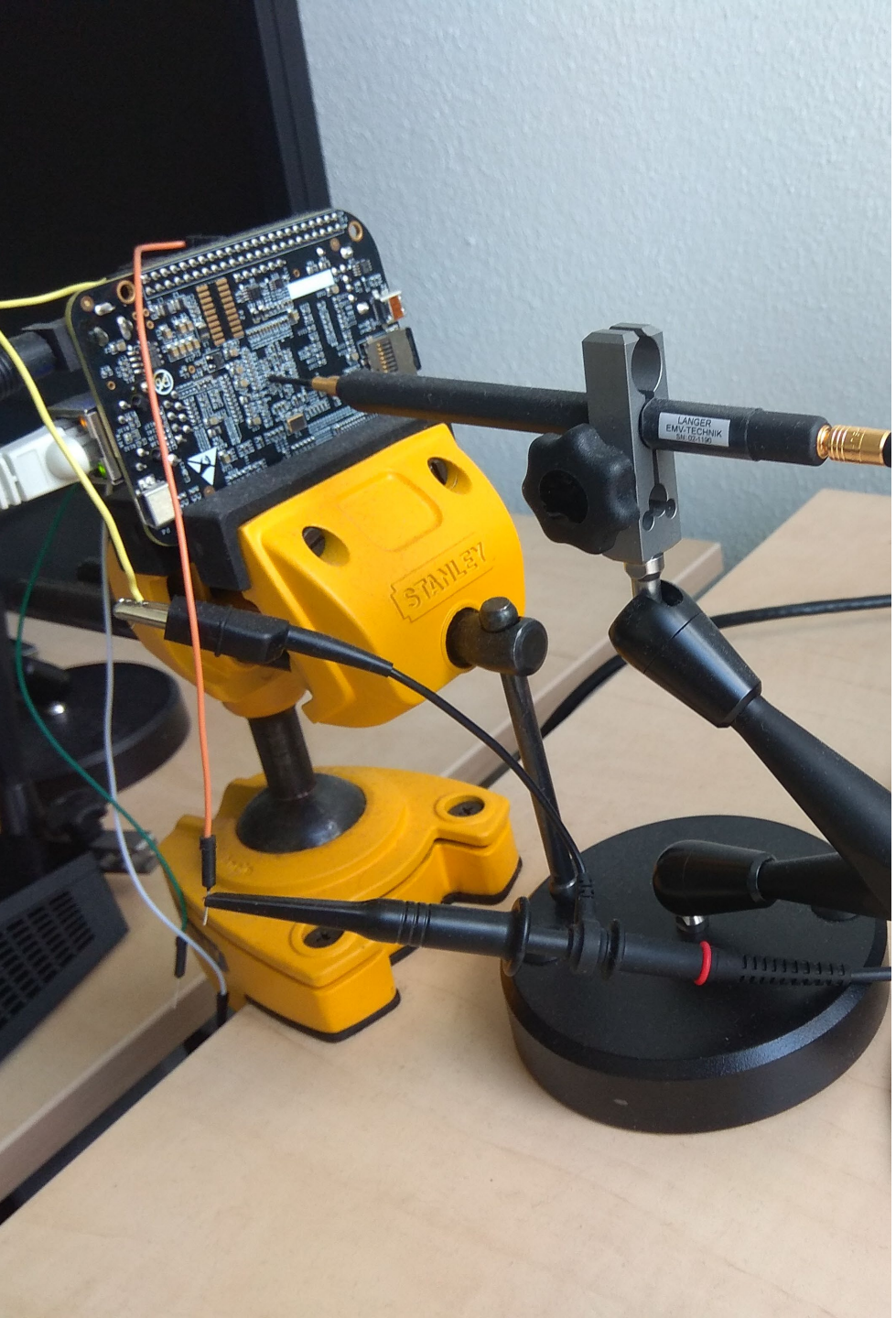
- $R_0 = R,\ R_1 = R + V, R_2 = -R$
- Convert R_0 , R_1 , R_2 to Montgomery format
- For i= t-1 to 0
  - $R_2 = 2R_2$ in permutat. γ_t
  - If k_i = 1

    R_0 = R_0 + R_1 and R_1 = $2R_1$ in permutat. γ_t

    else

    R_1 = R_0 + R_1 and R_0 = $2R_0$ in permutat. γ_t
- Integrity check: if i,k not modified and $R_0 + V = R_1$ then ret. $R_0 + R_2$ in permutat. γ_t

  else ret. random value



RNS implementation on BeagleBone

- C Software implementation on ARM Cortex A8
- RNS Montgomery multiplication
- Dedicated and Unified Group Law
- 5 different variations: unprotected, randomized scalar, random input point, random base permutations (LRA), random order of operations
- Broad and generic evaluation framework using ML classifiers and feature engineering

Evaluation Methodology

Proposed Model for Feature Processing

Preprocessing

ML Location Dependent Leakage

ML Data Dependent Leakage

Reduced Features Testing Dataset

Aligned Traces

ML Classification Model Training

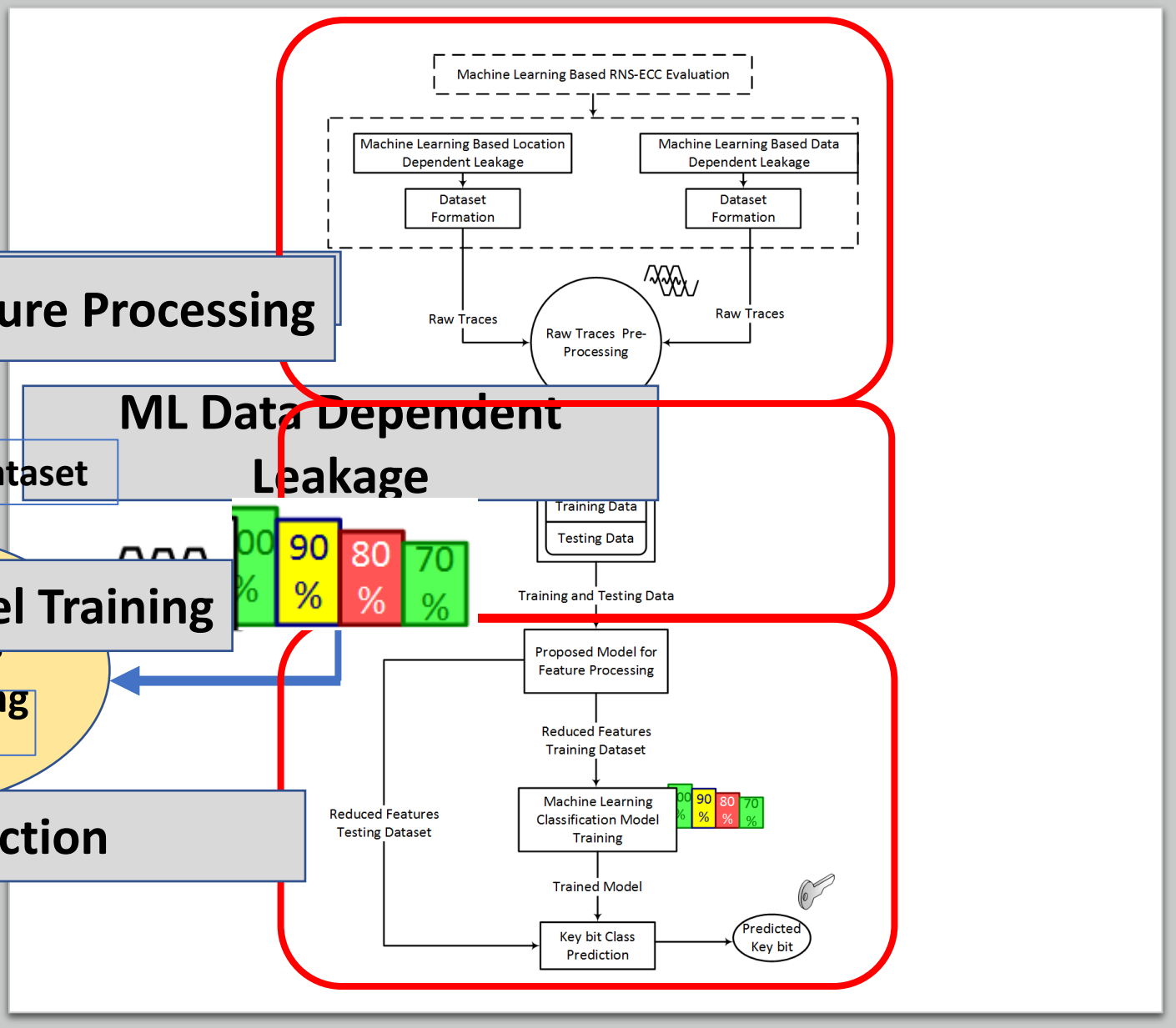
Data Splitting

- Training Data
- Testing Data

Trained Model

Key Bit Class Prediction

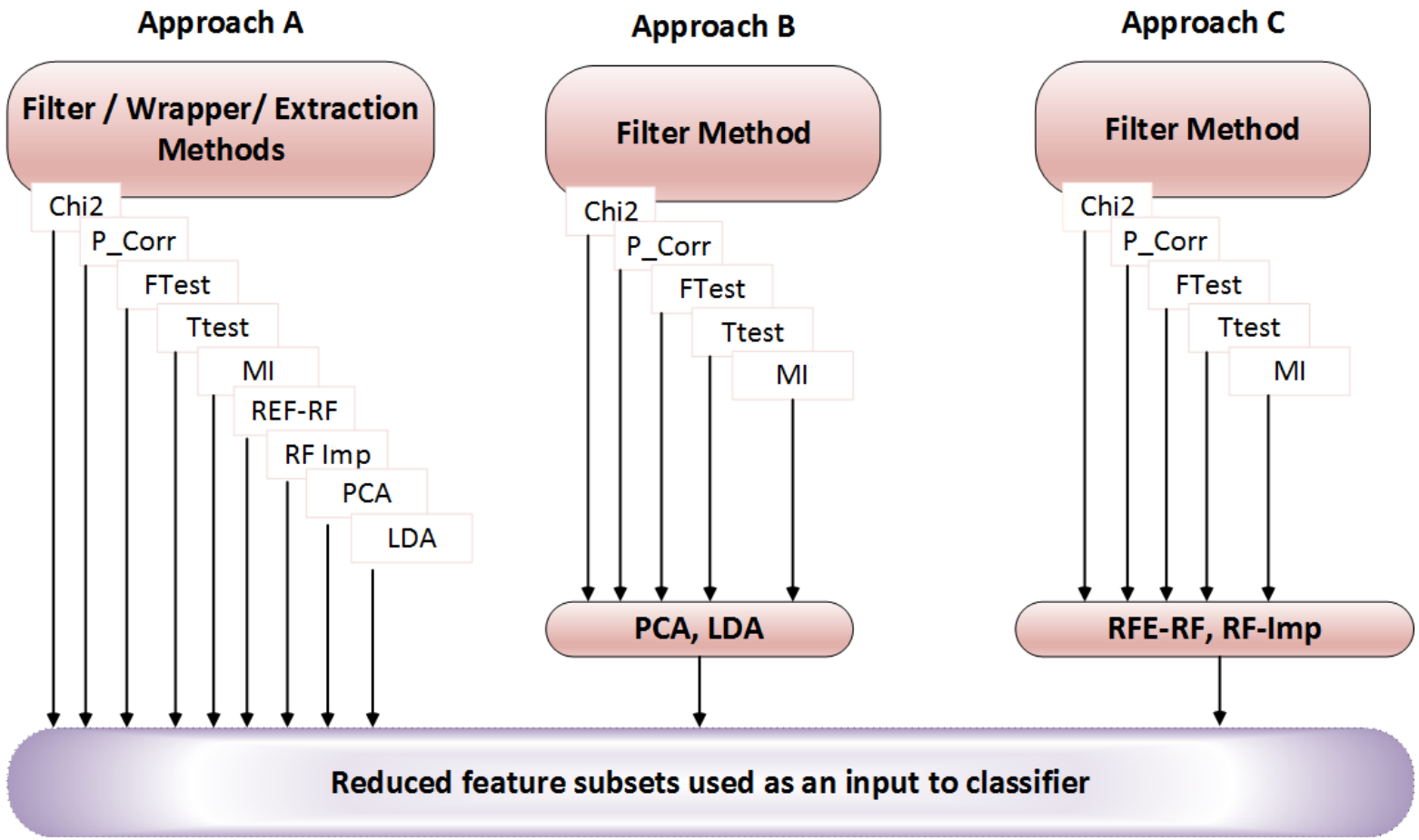
Training and Testing Data



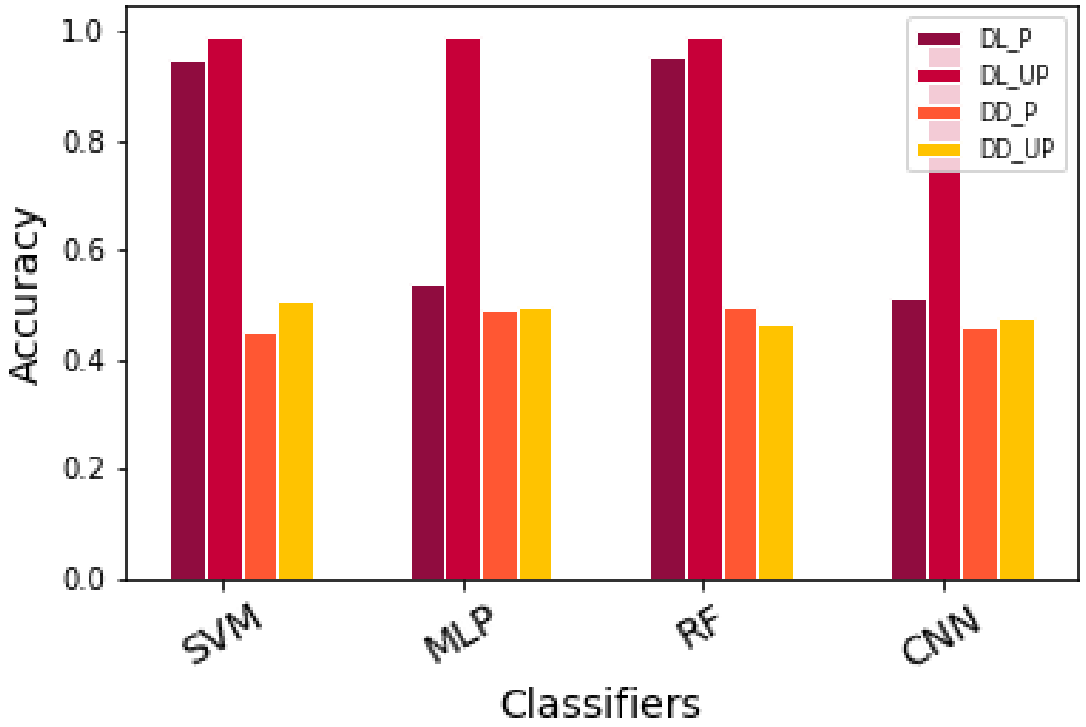
ML Classifiers

1. **Support Vector Machine (SVM)**
   N-dimensional data separated using hyperplane
2. **Random Forests (RF)**
   Data formed by aggregating the collection of decision trees
3. **Multi-Layer Perceptron (MLP)**
   Feed-forward artificial NN that uses back-propagation for learning
4. **Convolutional Neural Network (CNN)**
   CL performs convolution on the input features, using filters, to recognize patterns in the data

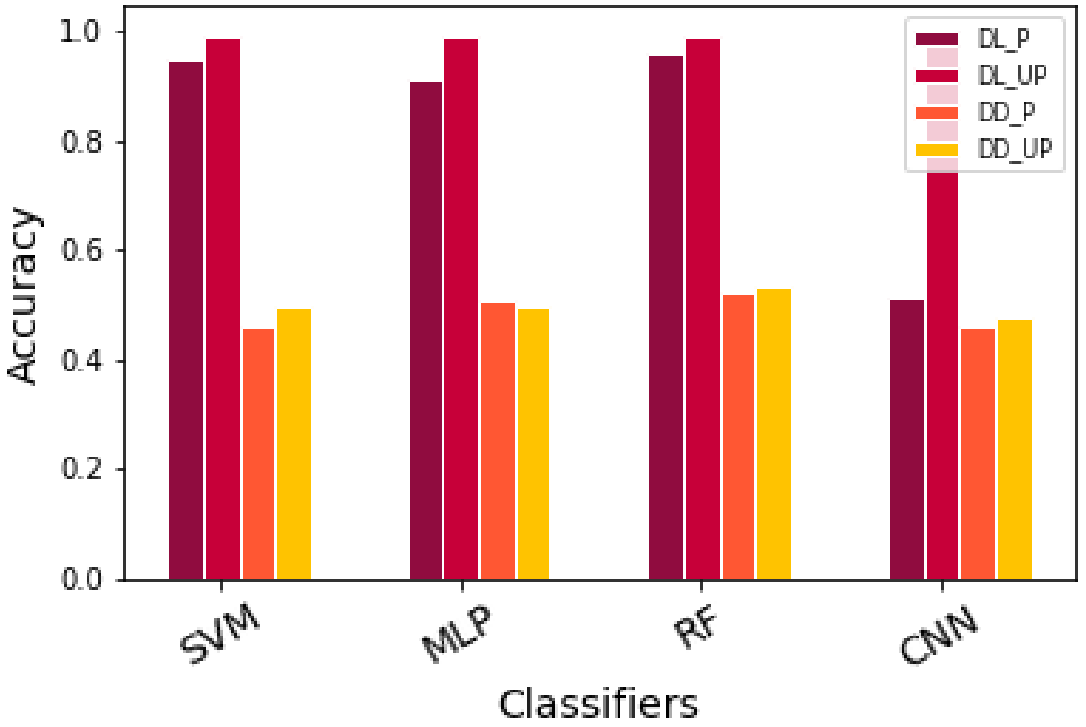
Feature Engineering Model



Accuracy of classifiers on raw features

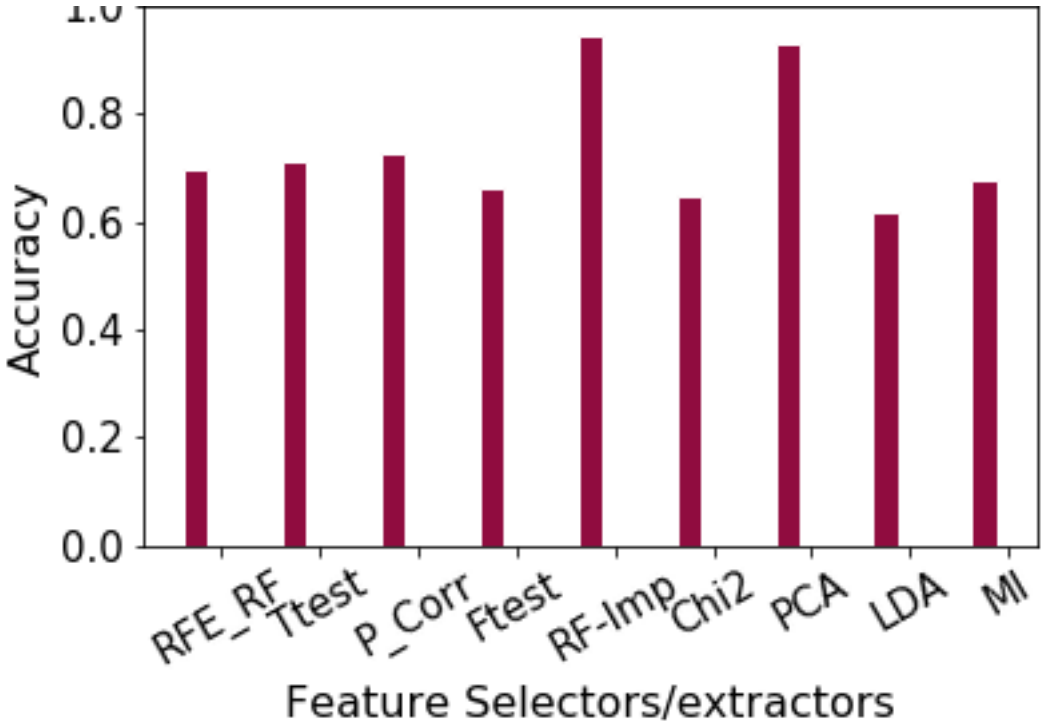


Full Length Traces used as Features for the Training Model

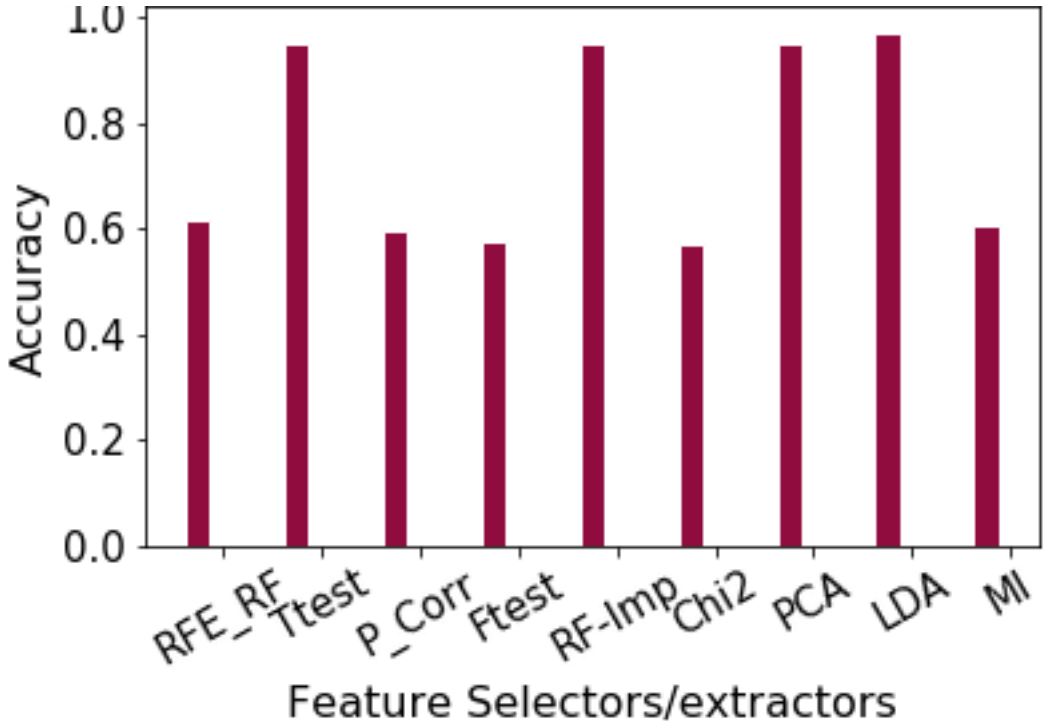


Reduced Features Dataset

Impact of Feature Engineering Techniques

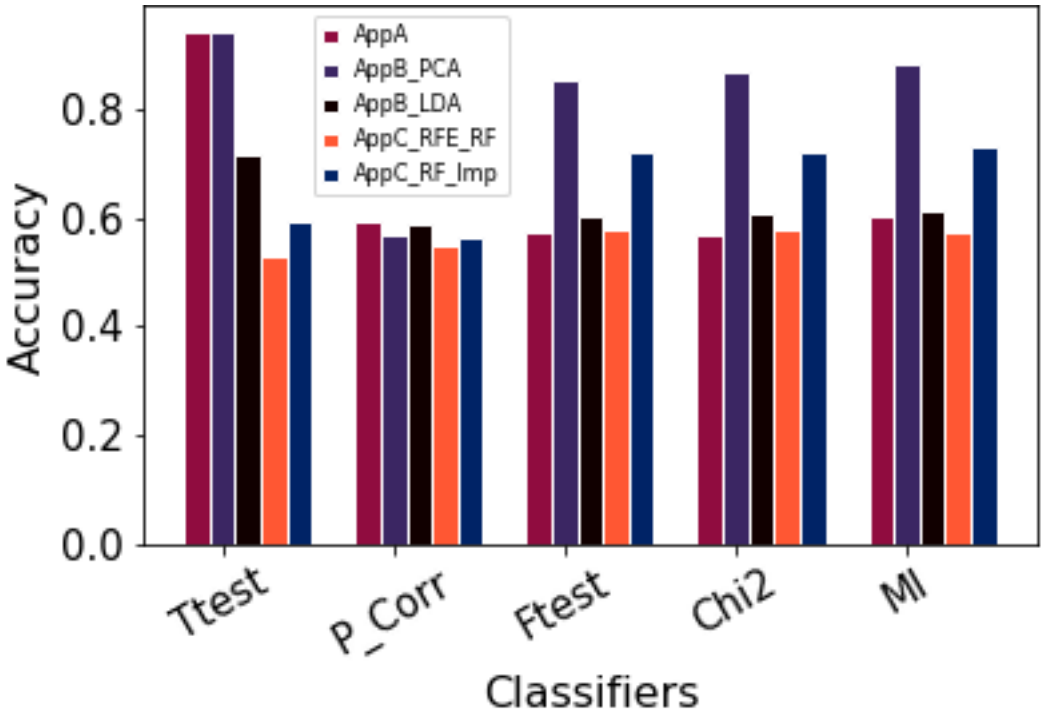


Trace Dataset with all samples

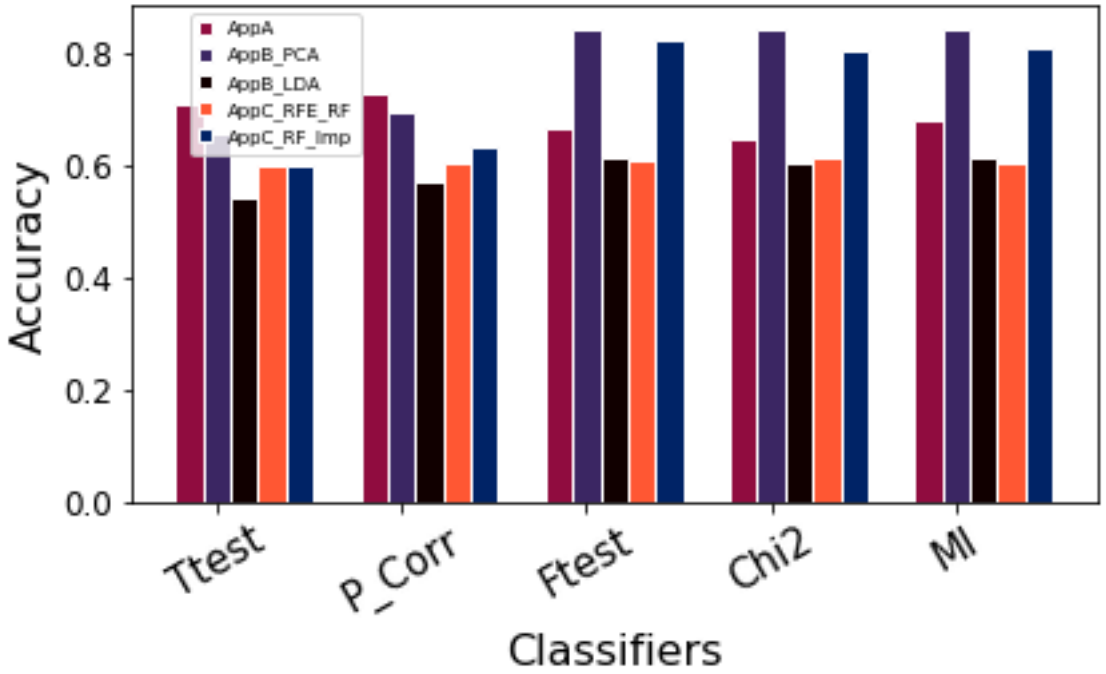


Traces with aligned reduced samples

Hybrid Feature Engineering Techniques



Trace Dataset with all samples



Traces with aligned reduced samples

Conclusions

- ML-based SCA on PKC are realistic and require less pre-processing compared to template attacks.
- The secret key can be recovered from unprotected and protected RNS ECC SM, using location-based attacks, with 99% and 95% ac curacy, respectively.
- Evaluated the effect of training a model with small dataset (order of 10k).
- RNS-ECC implementations showed resistance against ML-based data dependent attacks

THANK YOU FOR YOUR ATTENTION !