# Convolutional Neural Network based Side-Channel Attacks in Time-Frequency Representations

G. Yang[1,2] H. Li[1,2] J. Ming[1,2] Y. Zhou[1,2]

[1]State Key Laboratory of Information Security
Institute of Information Engineering
Chinese Academy of Sciences

[2]School of Cyber Security
University of Chinese Academy of Sciences

November 12, CARDIS 2018

# Outline

1. Introduction
   - Side-Channel Attacks (SCA)
   - Signal Representations in SCA
2. Related Work
   - Time-Frequency Representation of Signals
   - Deep Learning based SCA
3. Our Method
   - Main Idea
   - Leakages in Spectrograms
   - How to Use Convolutional Neural Networks (CNN) Exploit Leakages
4. Experiments
   - Setup of Spectrogram Parameters
   - Comparison of Attack Results
5. Conclusion

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline



中国科学院信息工程研究所 INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

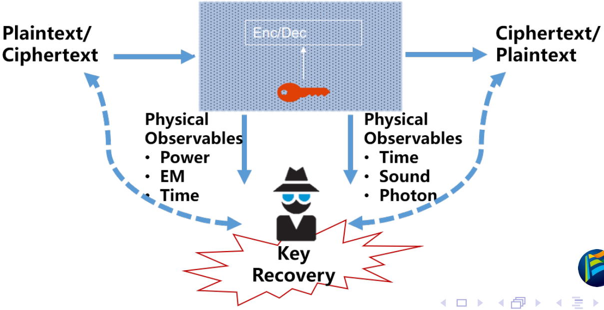
# Outline



中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

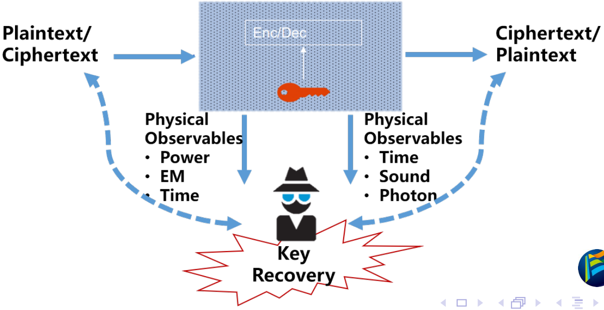
## Side-Channel Attacks (SCA)

- First introduced in 1996
- Exploit intermediate value correlated leakage (passively)
- Recover secret information of hardware implementations
- Of low cost, yet big threats to cryptographic implementations



## Side-Channel Attacks (SCA)

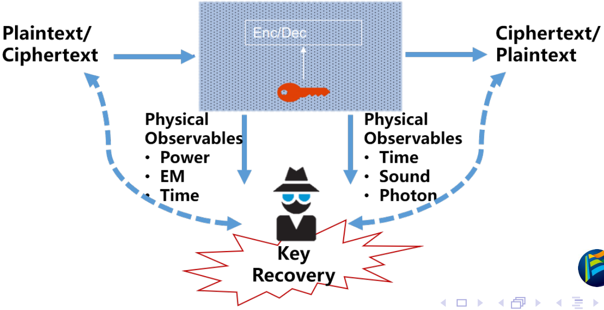
- First introduced in 1996
- Exploit intermediate value correlated leakage (passively)
- Recover secret information of hardware implementations
- Of low cost, yet big threats to cryptographic implementations




中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## Side-Channel Attacks (SCA)

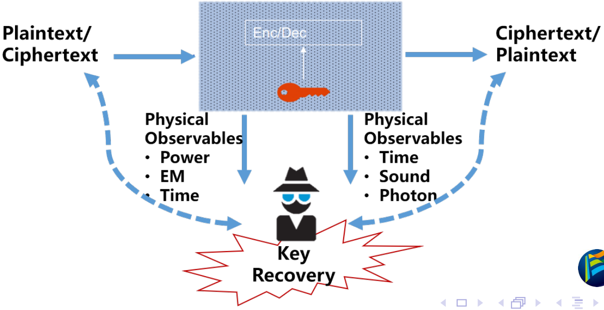
- First introduced in 1996
- Exploit intermediate value correlated leakage (passively)
- Recover secret information of hardware implementations
- Of low cost, yet big threats to cryptographic implementations




中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

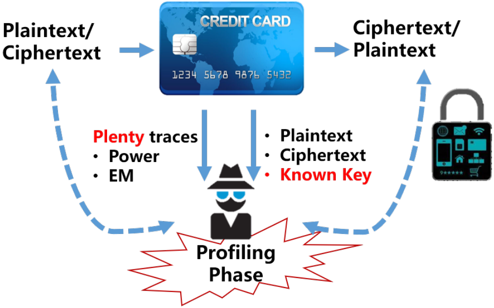
## Side-Channel Attacks (SCA)

- First introduced in 1996
- Exploit intermediate value correlated leakage (passively)
- Recover secret information of hardware implementations
- Of low cost, yet big threats to cryptographic implementations



## Profiled SCA

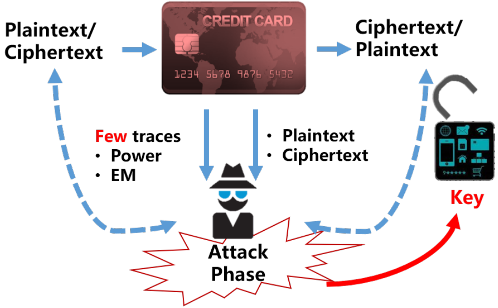
- **Profiling Phase:** perform leakage characterization with known ciphertext/plaintext and known keys
- **Attack Phase:** recover secrets within the target device using profiled leakage characterization




In this way, the WORST CASE SECURITY of cryptographic implementations is examined.

## Profiled SCA

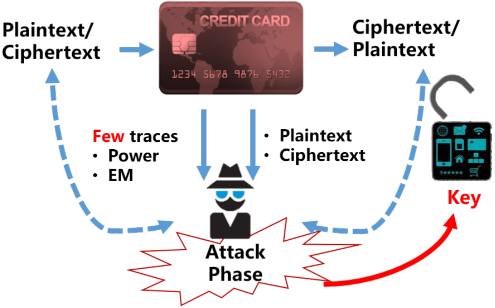
- **Profiling Phase:** perform leakage characterization with known ciphertext/plaintext and known keys
- **Attack Phase:** recover secrets within the target device using profiled leakage characterization




In this way, the WORST CASE SECURITY of cryptographic implementations is examined.

## Profiled SCA

- **Profiling Phase:** perform leakage characterization with known ciphertext/plaintext and known keys
- **Attack Phase:** recover secrets within the target device using profiled leakage characterization




In this way, the WORST CASE SECURITY of cryptographic implementations is examined.

## Notation

- **x:** side-channel leakage observables (e.g. Power, EM)
- **v:** sensitive variable ($v = f(p, k)$)

**Goal:** given **x**, estimate **v**

**Profiling:** Build models to accurately estimate prior probability $\Pr[\mathbf{x_i}|v = v_i]$

**Attack:** Calculate posterior probabilities among k guesses using Bayes theorem and Maximum Likelihood Criterion

$$d_k = \prod_{i=1}^{M} \Pr[v_i = f(t_i, k)|\mathbf{x} = \mathbf{x}_i]$$

$$= \prod_{i=1}^{M} \frac{\Pr[\mathbf{x} = \mathbf{x}_i|v_i = f(t_i, k)] \cdot \Pr[v_i = f(t_i, k)]}{\Pr[\mathbf{x} = \mathbf{x}_i]}$$

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## Notation

- **x:** side-channel leakage observables (e.g. Power, EM)
- **v:** sensitive variable ($v = f(p, k)$)

**Goal:** given **x**, estimate **v**

**Profiling:** Build models to accurately estimate prior probability $\Pr[\mathbf{x_i}|v = v_i]$

**Attack:** Calculate posterior probabilities among k guesses using Bayes theorem and Maximum Likelihood Criterion

$$d_k = \prod_{i=1}^{M} \Pr[v_i = f(t_i, k)|\mathbf{x} = \mathbf{x}_i]$$

$$= \prod_{i=1}^{M} \frac{\Pr[\mathbf{x} = \mathbf{x}_i|v_i = f(t_i, k)] \cdot \Pr[v_i = f(t_i, k)]}{\Pr[\mathbf{x} = \mathbf{x}_i]}$$

中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## Notation

- **x:** side-channel leakage observables (e.g. Power, EM)
- **v:** sensitive variable ($v = f(p, k)$)

**Goal:** given **x**, estimate **v**

**Profiling:** Build models to accurately estimate prior probability $\Pr[\mathbf{x_i} | v = v_i]$

**Attack:** Calculate posterior probabilities among k guesses using Bayes theorem and Maximum Likelihood Criterion

$$d_k = \prod_{i=1}^{M} \Pr[v_i = f(t_i, k) | \mathbf{x} = \mathbf{x}_i]$$

$$= \prod_{i=1}^{M} \frac{\Pr[\mathbf{x} = \mathbf{x}_i | v_i = f(t_i, k)] \cdot \Pr[v_i = f(t_i, k)]}{\Pr[\mathbf{x} = \mathbf{x}_i]}$$

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## State-of-the-art Profiled Attack Techniques:

- Template Attacks and Stochastic Model
- Machine learning (e.g. SVM, Random Forest) and deep learning (e.g. CNN, MLP) based attacks

### Template Attacks

**Pros:**

- Theoretically perfect
- Robust and explainable

**Cons:**

- Dependency of preprocessing
- Numerical problems
- Curse of dimensionality

### Deep Learning Techniques

**Pros:**

- ~~Dependency of preprocessing~~
- ~~Numerical problems~~
- ~~Curse of dimensionality~~
- High-order analysis

**Cons:**

- More traces needed!

## State-of-the-art Profiled Attack Techniques:

- Template Attacks and Stochastic Model
- Machine learning (e.g. SVM, Random Forest) and deep learning (e.g. CNN, MLP) based attacks

### Template Attacks

**Pros:**

- Theoretically perfect
- Robust and explainable

**Cons:**

- Dependency of preprocessing
- Numerical problems
- Curse of dimensionality

### Deep Learning Techniques

**Pros:**

- ~~Dependency of preprocessing~~
- ~~Numerical problems~~
- ~~Curse of dimensionality~~
- High-order analysis

**Cons:**

- More traces needed!

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## State-of-the-art Profiled Attack Techniques:

- Template Attacks and Stochastic Model
- Machine learning (e.g. SVM, Random Forest) and deep learning (e.g. CNN, MLP) based attacks

**Template Attacks**

**Pros:**

- Theoretically perfect
- Robust and explainable

**Cons:**

- Dependency of preprocessing
- Numerical problems
- Curse of dimensionality

**Deep Learning Techniques**

**Pros:**

- ~~Dependency of preprocessing~~
- ~~Numerical problems~~
- ~~Curse of dimensionality~~
- High-order analysis

**Cons:**

- More traces needed!

## State-of-the-art Profiled Attack Techniques:

- Template Attacks and Stochastic Model
- Machine learning (e.g. SVM, Random Forest) and deep learning (e.g. CNN, MLP) based attacks

**Template Attacks**

**Pros:**

- Theoretically perfect
- Robust and explainable

**Cons:**

- Dependency of preprocessing
- Numerical problems
- Curse of dimensionality

**Deep Learning Techniques**

**Pros:**

- ~~Dependency of preprocessing~~
- ~~Numerical problems~~
- ~~Curse of dimensionality~~
- High-order analysis

**Cons:**

- More traces needed!

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## State-of-the-art Profiled Attack Techniques:

- Template Attacks and Stochastic Model
- Machine learning (e.g. SVM, Random Forest) and deep learning (e.g. CNN, MLP) based attacks

**Template Attacks**

**Pros:**

- Theoretically perfect
- Robust and explainable

**Cons:**

- Dependency of preprocessing
- Numerical problems
- Curse of dimensionality

**Deep Learning Techniques**

**Pros:**

- ~~Dependency of preprocessing~~
- ~~Numerical problems~~
- ~~Curse of dimensionality~~
- High-order analysis

**Cons:**

- More traces needed!

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline

1. Introduction
   - Side-Channel Attacks (SCA)
   - Signal Representations in SCA
2. Related Work
   - Time-Frequency Representation of Signals
   - Deep Learning based SCA
3. Our Method
   - Main Idea
   - Leakages in Spectrograms
   - How to Use Convolutional Neural Networks (CNN) Exploit Leakages
4. Experiments
   - Setup of Spectrogram Parameters
   - Comparison of Attack Results
5. Conclusion

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

**SCA in time domain**

- Easy to deploy
- On raw traces, no information loss in preprocessing ideally

**SCA in frequency domain**

- Fourier transform needed
- Suitable for misaligned traces
- Time information is lost

**In practice**, most profiled attacks are performed on time domain, in which some frequency related leakage may lose...

**SCA in time domain**

- Easy to deploy
- On raw traces, no information loss in preprocessing ideally

**SCA in frequency domain**

- Fourier transform needed
- Suitable for misaligned traces
- Time information is lost

**In practice**, most profiled attacks are performed on time domain, in which some frequency related leakage may lose...

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Related Work

## Time-Frequency Representation of Signals

Spectrogram is widely used for signal processing, e.g. speech processing, sonar and radar.

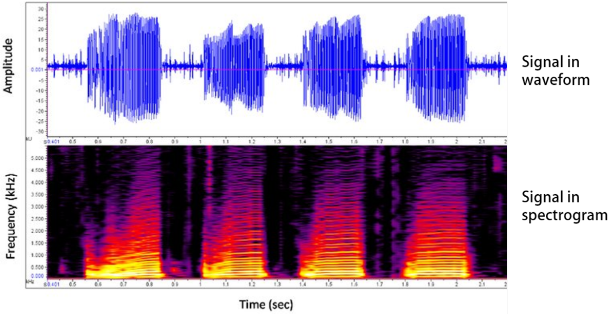



Figure: A boat whistle signal and its time-frequency representation

In the field of SCA, short-time Fourier transform or Wavelet transform is used as preprocessing method in non-profiled attacks (e.g. CPA).

# Outline

**A Review of deep learning based side-channel attacks...**

- [MPP16] First using Convolutional Neural Networks (CNN) into SCA
- [CDP17] Introduction of CNN to analyse mis-alignment traces / Providing data augmentation methods
- [Pro+18] A detailed study of deep learning hyper-parameters for SCA

**These works mainly focus SCA on time domain, what about the leakage information in frequency domain?**

### Our Purpose

Following the line of deep learning based attacks,

- Solve masking/mis-alignment problems [MPP16; CDP17; Pro+18]

and **bring new features:**

- **Time-frequency analysis (ours)**

**A Review of deep learning based side-channel attacks...**

- [MPP16] First using Convolutional Neural Networks (CNN) into SCA
- [CDP17] Introduction of CNN to analyse mis-alignment traces / Providing data augmentation methods
- [Pro+18] A detailed study of deep learning hyper-parameters for SCA

**These works mainly focus SCA on time domain, what about the leakage information in frequency domain?**

### Our Purpose

Following the line of deep learning based attacks,

- Solve masking/mis-alignment problems [MPP16; CDP17; Pro+18]

and **bring new features:**

- **Time-frequency analysis (ours)**

**A Review of deep learning based side-channel attacks...**

- [MPP16] First using Convolutional Neural Networks (CNN) into SCA
- [CDP17] Introduction of CNN to analyse mis-alignment traces / Providing data augmentation methods
- [Pro+18] A detailed study of deep learning hyper-parameters for SCA

**These works mainly focus SCA on time domain, what about the leakage information in frequency domain?**

### Our Purpose

Following the line of deep learning based attacks,

- Solve masking/mis-alignment problems [MPP16; CDP17; Pro+18]

and **bring new features:**

- **Time-frequency analysis (ours)**

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Our Method

## Main Idea

- We use short-time Fourier transform (STFT) to generate 2D spectrograms, instead of 1D traces, as the input of profiled attacks.
- We intend to make the most of CNN to exploit local time-frequency leakage information, just like recognizing dogs in an image.

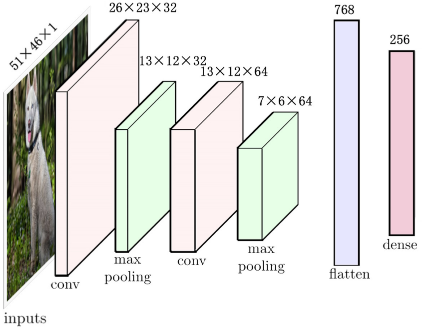



Figure: Classification problem of dogs

# Our Method

## Main Idea

- We use short-time Fourier transform (STFT) to generate 2D spectrograms, instead of 1D traces, as the input of profiled attacks.
- We intend to make the most of CNN to exploit local time-frequency leakage information, just like recognizing dogs in an image.

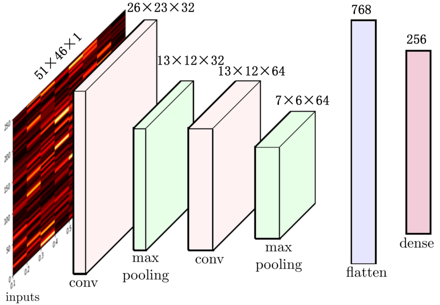



Figure: Classification problem of spectrograms

Let's first see what is spectrogram and how's the leakage in spectrograms. Then I'll introduce how we ultilize 2D CNN to exploit the local time-frequency leakages in spectrograms.

# Outline



中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

### Definition

A **spectrogram** is a visual way of representing the signal strength of a signal over time at various frequencies present in a particular waveform.

- It's the magnitude of STFT
- Two axes: time and frequency. The value is magnitude of a particular frequency at a particular time
- Usually shown in the form of a heatmap

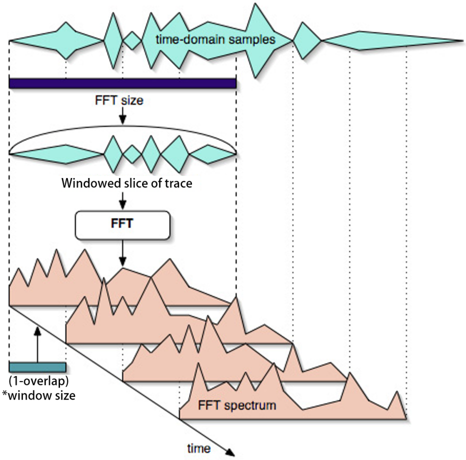
**How do we turn traces into spectrograms?**

**Step 1: Perform short-time Fourier transform on traces**

$$\mathrm{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-mH]e^{-j\omega n}$$

**Step 2: Calculate the magnitude of STFT**

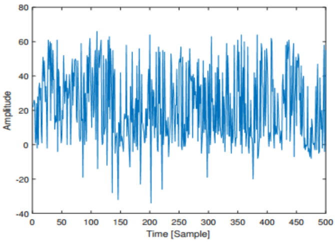
$$\mathrm{spectrogram}\{x[n]\}(m,\omega) \equiv |X(m,\omega)|$$



**How do we turn traces into spectrograms?**

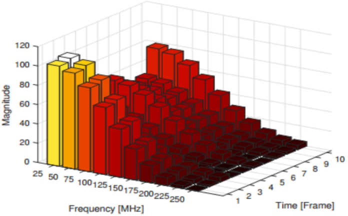
**Step 1: Perform short-time Fourier transform on traces**

$$\mathrm{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-mH]e^{-j\omega n}$$




**Step 2: Calculate the magnitude of STFT**

$$\mathrm{spectrogram}\{x[n]\}(m,\omega) \equiv |X(m,\omega)|^2$$

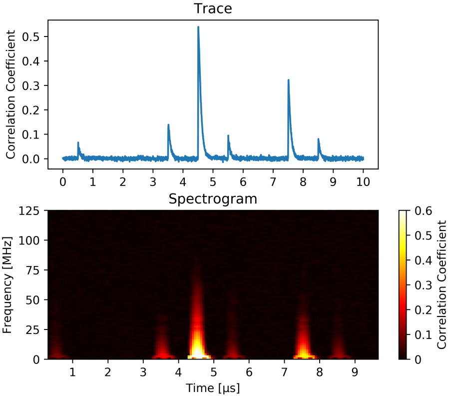


# Our Method

## Leakages in Spectrograms

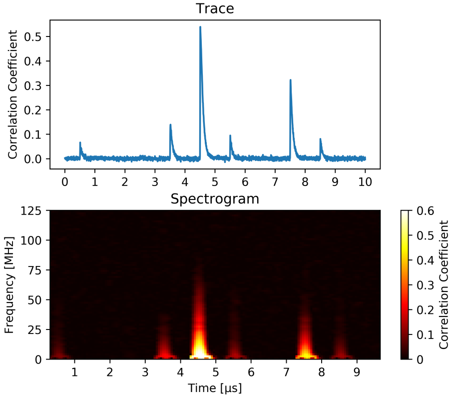
We can perform leakage detection on spectrograms and show the results in heatmaps.

- **Pearson Correlation Coefficient:** $\rho_{x,v} = \frac{\mathrm{cov}(x,v)}{\sigma_x \cdot \sigma_v}$
  - Trace: correlation coefficient peak value is 0.539
  - Spectrogram: correlation coefficient peak value is 0.626
- **Signal Noise Ratio (SNR):** $snr_{x,v} = \mathrm{Var}[E[x|v]]/E[\mathrm{Var}[x|v]]$
  - Trace: SNR peak value is 1.781
  - Spectrogram: SNR peak value is 5.878



We can perform leakage detection on spectrograms and show the results in heatmaps.

- **Pearson Correlation Coefficient:** $\rho_{x,v} = \frac{\text{cov}(x,v)}{\sigma_x \cdot \sigma_v}$
  - Trace: correlation coefficient peak value is 0.539
  - Spectrogram: correlation coefficient peak value is 0.626
- **Signal Noise Ratio (SNR):** $snr_{x,v} = \text{Var}[E[x|v]]/E[\text{Var}[x|v]]$
  - Trace: SNR peak value is 1.781
  - Spectrogram: SNR peak value is 5.878

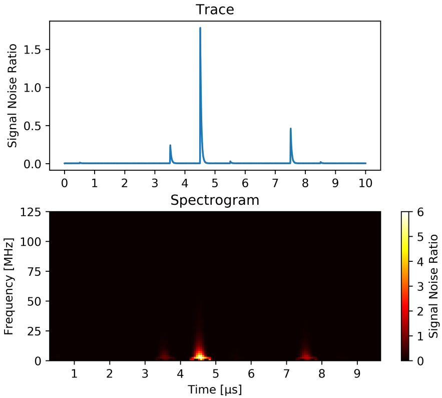


# Our Method

## Leakages in Spectrograms

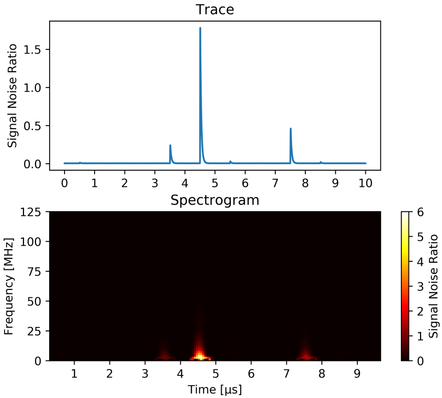
We can perform leakage detection on spectrograms and show the results in heatmaps.

- **Pearson Correlation Coefficient:** $\rho_{x,v} = \frac{\mathrm{cov}(x,v)}{\sigma_x \cdot \sigma_v}$
  - Trace: correlation coefficient peak value is 0.539
  - Spectrogram: correlation coefficient peak value is 0.626
- **Signal Noise Ratio (SNR):** $snr_{x,v} = \mathrm{Var}[\mathrm{E}[x|v]]/\mathrm{E}[\mathrm{Var}[x|v]]$
  - Trace: SNR peak value is 1.781
  - Spectrogram: SNR peak value is 5.878



We can perform leakage detection on spectrograms and show the results in heatmaps.

- **Pearson Correlation Coefficient:** $\rho_{x,v} = \frac{\text{cov}(x,v)}{\sigma_x \cdot \sigma_v}$
  - Trace: correlation coefficient peak value is 0.539
  - Spectrogram: correlation coefficient peak value is 0.626
- **Signal Noise Ratio (SNR):** $snr_{x,v} = \text{Var}[\text{E}[x|v]]/\text{E}[\text{Var}[x|v]]$
  - Trace: SNR peak value is 1.781
  - Spectrogram: SNR peak value is 5.878




INSTITUTE OF INFORMATION ENGINEERING,CAS

# Our Method
## Leakages in Spectrograms

**POI appear in clusters and have certain 2D pattern features.** Better find a new way to analyse the feature of this pattern, otherwise POI selection would destroy the spacial relationship.

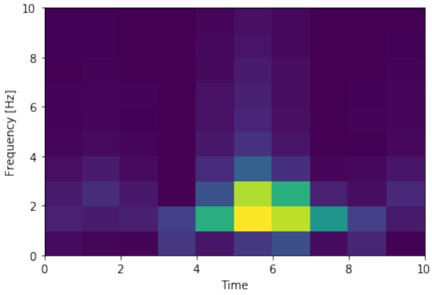



Figure: Enlarged partial detail of POI region in spectrogram

中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Outline

**A 2D CNN is composed of two parts:**

- Feature extraction: convolutional layer, pooling layer
- Classification: fully connected layer

The former part is used to extract local time-frequency leakage information, and the latter part is used to make classification.

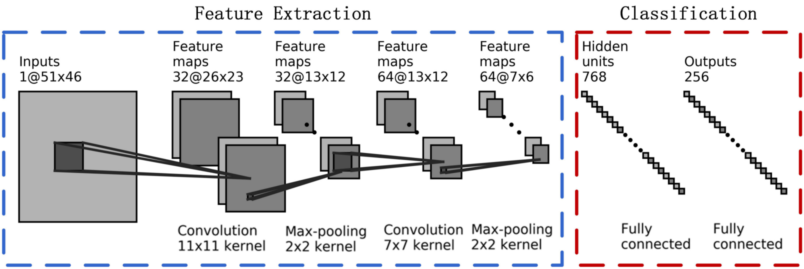



Figure: Basic CNN architecture

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## Convolutional Layer

It is locally connected with shared weights in learnable kernels. It helps recognizing local time-frequency patterns.

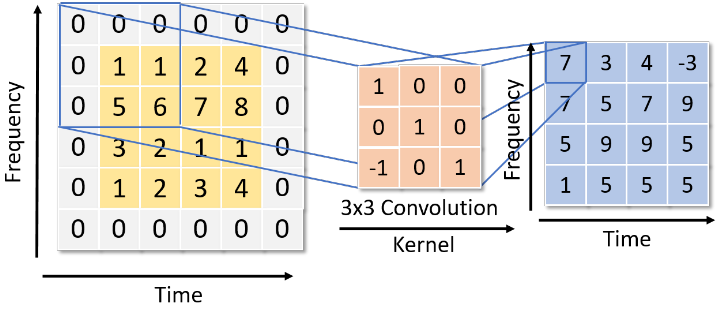



Figure: Convolution details

# Our Method

## How to Use Convolutional Neural Networks (CNN) Exploit Leakages

### Pooling Layer

It performs the downsampled operations to extract time-frequency features and discard unnecessary details.

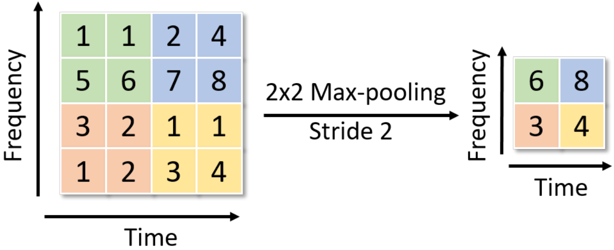



Figure: Max-pooling details

## Fully Connected Layer

Each neural is connected to the next layer with trainable weights. It helps combining features and making classification.

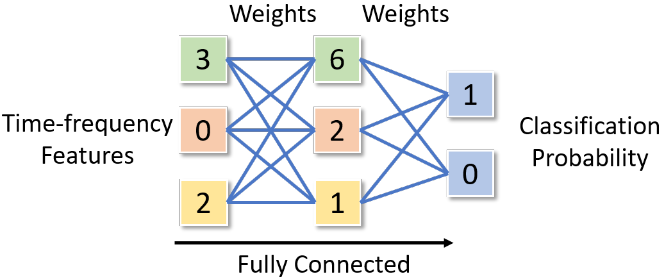



Figure: Fully connected layer details

# Outline



中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

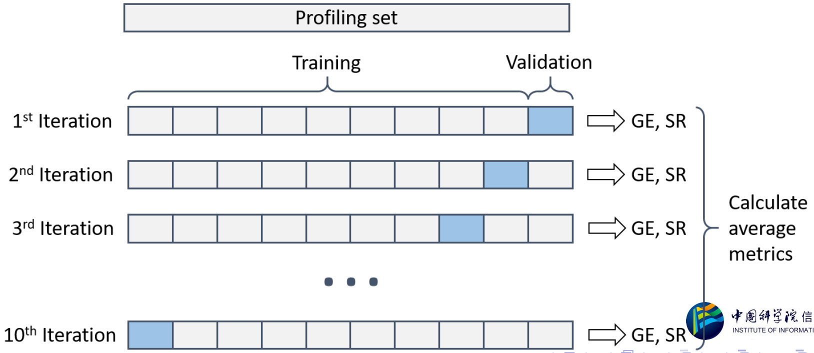
## Spectrogram Parameters

- Window type: Hanning window
- Window overlap: 90%
- Window size:
  - Small window size: coarse frequency resolution, but good time resolution
  - Large window size: good frequency resolution, but coarse time resolution

To find proper STFT window size, 10-fold cross validation is performed...

## 10-Fold Cross Validation to Evaluate the STFT Window Size

- Split profiling set, 9 folds as training set, 1 fold as validation set
- Iteratively train 10 times, calculate GE, SR on each validation set
- Calculate average metrics



**Experiments on 3 public datasets**

- **DPA contest V4.1 (DPAv4.1)**
  - Atmel ATMega-163 smart-card, AES-256
  - About 125 sample points per clock
  - Sbox out XOR mask, $V = \text{Sbox}[P \oplus k^*] \oplus M$
  - Profiling set: 9000, attack set: 1000
- **Grizzly**
  - 8-bit CPU Atmel XMEGA 256 A3U
  - About 1000 sample points per clock
  - Given label V, could be seen as Sbox out
  - Profiling set: 51200, attack set: 10000
- **DPA contest V2 (DPAv2)**
  - SASEBO GII FPGA, AES-128
  - About 213 sample points per clock
  - Sbox in XOR Sbox out, $V = \text{Sbox}^{-1}[C_1 \oplus k^*] \oplus C_2$
  - Profiling set: 90000, attack set: 10000

中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## DPAv4.1 Window Size Cross Validation Results

- **Time:** 3 hours (3 minutes per single training)
- **Configuration:** Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz CPU, 2 NVIDIA Titan Xp GPUs

| Window@percentage | | Spc size | Loss | Acc | Top3 Acc | GE<1 | SR>80% |
|---|---|---|---|---|---|---|---|
| | 8@1/16 | (4,494) | 0.159 | 95.3% | 99.6% | 1 | 1 |
| | 16@1/8 | (8,243) | 0.168 | 94.9% | 99.7% | 1 | 1 |
| DPAv4.1 | 32@1/4 | (16,181) | 0.153 | 95.2% | 99.7% | 1 | 1 |
| | 64@1/2 | (32,63) | **0.142** | **95.9%** | **99.7%** | 1 | 1 |
| | 125@1 | (63,29) | 0.199 | 94.1% | 99.6% | 1 | 1 |
| | 187@3/2 | (94,17) | 0.195 | 94.5% | 99.5% | 1 | 1 |

Best STFT window size is 64 (1/2 of a clock) points.

## Grizzly Window Size Cross Validation Results

- **Time:** 6 hours (6 minutes per single training)
- **Configuration:** Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz CPU, 2 NVIDIA Titan Xp GPUs

| Window@percentage | | Spc size | Loss | Acc | Top3 Acc | GE<1 | SR>80% |
|---|---|---|---|---|---|---|---|
| | 62@1/16 | (32,349) | 4.08 | 6.56% | 16.86% | 5 | 5 |
| | 125@1/8 | (63,183) | **3.74** | **8.49%** | **21.28%** | **3** | **4** |
| Grizzly | 250@1/4 | (126,91) | 3.76 | 8.28% | 21.07% | 3 | 4 |
| | 500@1/2 | (251,41) | 5.00 | 2.95% | 7.40% | >10 | >10 |
| | 1000@1 | (501,16) | 5.51 | 0.5% | 1.53% | >10 | >10 |

Best STFT window size is 125 (1/8 of a clock) points.

## DPAv2 Window Size Cross Validation Results

- **Time:** 8 hours (8 minutes per single training)
- **Configuration:** Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz CPU, 2 NVIDIA Titan Xp GPUs

| Window@percentage | | Spc size | Loss | Acc | Top3 Acc | GE<1 | SR>80% |
|---|---|---|---|---|---|---|---|
| DPAv2 | 12@1/16 | (6,495) | 5.544 | 0.43% | 1.29% | >1500 | >1500 |
| | 25@1/8 | (12,326) | 5.544 | 0.43% | 1.30% | >1500 | >1500 |
| | 50@1/4 | (25,191) | 5.536 | 0.62% | 1.63% | 750 | 750 |
| | 100@1/2 | (50,91) | **5.536** | **0.65%** | **1.67%** | **700** | **700** |
| | 200@1 | (100,41) | 5.538 | 0.60% | 1.58% | 950 | 900 |
| | 300@3/2 | (300,48) | 5.538 | 0.63% | 1.60% | 950 | 950 |

Best STFT window size is 100 (1/2 of a clock) points.

中国科学院 信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

## Experimental Conclusion

- Choice of imbalanced spectrogram size usually results in training failure
- The window size 64, 128, 256 suits most case in our experiments

## An Example on Grizzly

- Trace length 2500, STFT window size 1000
- Spectrogram size $501 \times 16$
- After 4 CONV and Pooling layers
- Feature map size $32 \times 1$ (redundant frequency information but exhausted temporal information)

# Outline

We compare the efficiency of TA and CNN based attacks on traces and spectrograms.

**Targets**

- **DPAv4.1**, 9000 traces for profiling, 1000 traces for attack
- **Grizzly**, 51200 traces for profiling, 10000 traces for attack
- **DPAv2**, 90000 traces for profilng, 10000 traces for attack

We compare the efficiency of TA and CNN based attacks on traces and spectrograms.

**Targets**

- **DPAv4.1**, 9000 traces for profiling, 1000 traces for attack
- **Grizzly**, 51200 traces for profiling, 10000 traces for attack
- **DPAv2**, 90000 traces for profilng, 10000 traces for attack

**Profiling Methods**

- **CNN:** VGG-like architecture (detailed in paper)
- **ETA:** Efficient Template Attack with POI selection
- **PCA-ETA:** Efficient Template Attack with PCA dimension reduction

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

We compare the efficiency of TA and CNN based attacks on traces and spectrograms.

**Targets**

- **DPAv4.1**, 9000 traces for profiling, 1000 traces for attack
- **Grizzly**, 51200 traces for profiling, 10000 traces for attack
- **DPAv2**, 90000 traces for profilng, 10000 traces for attack

**Profiling Methods**

- **CNN:** VGG-like architecture (detailed in paper)
- **ETA:** Efficient Template Attack with POI selection
- **PCA-ETA:** Efficient Template Attack with PCA dimension reduction

**Signal Representations**

- **Trc:** 1D raw trace
- **Spc:** 2D spectrogram

Table: Attack results of our method and baseline methods.

| Method | | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

Table: Attack results of our method and baseline methods.

| Method | | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

Table: Attack results of our method and baseline methods.

| | Method | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

# Experiments

## Comparison of Attack Results

Table: Attack results of our method and baseline methods.

| Method | | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

Table: Attack results of our method and baseline methods.

| Method | | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

Table: Attack results of our method and baseline methods.

| Method | | DPAv4.1 | | | Grizzly | | | DPAv2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 | Acc | GE<1 | SR>0.8 |
| Spc | CNN | 95.5% | **1** | **1** | 8.47% | **3** | **4** | **0.82%** | **400** | **550** |
| | ETA,5poi | 15.0% | 4 | 3 | 2.46% | 7 | 5 | 0.67% | 600 | 550 |
| | ETA,25poi | 58.4% | 2 | 2 | 2.85% | 6 | 6 | 0.61% | 650 | 750 |
| | ETA,50poi | 82.5% | 1 | 1 | 3.64% | 5 | 5 | 0.65% | 1000 | 1050 |
| | PCA-ETA | 82.5% | 1 | 1 | 5.75% | 5 | 4 | 0.59% | 650 | 650 |
| Trc | CNN | **96.5%** | 1 | 1 | **9.52%** | **3** | **4** | 0.63% | 750 | 650 |
| | ETA,5poi | 1.9% | 9 | 7 | 2.08% | 8 | 7 | 0.59% | 1500 | 1500 |
| | ETA,25poi | 32.1% | 2 | 2 | 2.76% | 7 | 6 | 0.61% | 950 | 1000 |
| | ETA,50poi | 63.5% | 2 | 2 | 2.59% | 7 | 6 | 0.57% | 750 | 850 |
| | PCA-ETA | 86.9% | 1 | 1 | 4.48% | 6 | 5 | 0.60% | 850 | 750 |

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Conclusion

- Leakage in time-frequency 2D patterns can be ultilized simutaneously with the help of 2D CNN.
- 2D CNN extracts features by recognizing local time-frequency pattern (natural tool to block irrelevant time-frequency area without POI selection). In contrast, TA is unable to process spacial relations.
- Proper STFT window size helps training 2D CNN model.
- CNN based SCA in time-frequency representations provides an alternative way for deep learning based attacks.

- Future works
  - The performance of 2D CNN based profiled attacks in the presence of masking and hiding?

# Conclusion

- Leakage in time-frequency 2D patterns can be ultilized simutaneously with the help of 2D CNN.
- 2D CNN extracts features by recognizing local time-frequency pattern (natural tool to block irrelevant time-frequency area without POI selection). In contrast, TA is unable to process spacial relations.
- Proper STFT window size helps training 2D CNN model.
- CNN based SCA in time-frequency representations provides an alternative way for deep learning based attacks.

- Future works
  - The performance of 2D CNN based profiled attacks in the presence of masking and hiding?

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Conclusion

- Leakage in time-frequency 2D patterns can be ultilized simutaneously with the help of 2D CNN.
- 2D CNN extracts features by recognizing local time-frequency pattern (natural tool to block irrelevant time-frequency area without POI selection). In contrast, TA is unable to process spacial relations.
- Proper STFT window size helps training 2D CNN model.
- CNN based SCA in time-frequency representations provides an alternative way for deep learning based attacks.

- Future works
  - The performance of 2D CNN based profiled attacks in the presence of masking and hiding?

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Conclusion

- Leakage in time-frequency 2D patterns can be ultilized simutaneously with the help of 2D CNN.
- 2D CNN extracts features by recognizing local time-frequency pattern (natural tool to block irrelevant time-frequency area without POI selection). In contrast, TA is unable to process spacial relations.
- Proper STFT window size helps training 2D CNN model.
- CNN based SCA in time-frequency representations provides an alternative way for deep learning based attacks.
- Future works
  - The performance of 2D CNN based profiled attacks in the presence of masking and hiding?

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Conclusion

- Leakage in time-frequency 2D patterns can be ultilized simutaneously with the help of 2D CNN.
- 2D CNN extracts features by recognizing local time-frequency pattern (natural tool to block irrelevant time-frequency area without POI selection). In contrast, TA is unable to process spacial relations.
- Proper STFT window size helps training 2D CNN model.
- CNN based SCA in time-frequency representations provides an alternative way for deep learning based attacks.
- Future works
  - The performance of 2D CNN based profiled attacks in the presence of masking and hiding?

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

# Questions

Thank you! Any questions?

# Further Reading


Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. "Breaking cryptographic implementations using deep learning techniques". In: SPACE. Springer. 2016, pp. 3–26.

Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. "Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures". In: CHES. Springer. 2017, pp. 45–68.

Emmanuel Prouff et al. "Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database". In: IACR Cryptology ePrint Archive 2018 (2018), p. 53.