Technical Investigation Report

Side-Channel Analysis Pipeline for 3DES Smartcards

Data Compatibility and Feasibility Assessment

Khaled Koubaa

February 20, 2026

Abstract

This report presents the findings of a comprehensive technical investigation into the Side-Channel Analysis (SCA) pipeline designed for Triple-DES (3DES) key recovery from EMV smartcards. The investigation revealed fundamental data limitations that prevent reliable key recovery from blind traces. Specifically, we identified insufficient S-Box output coverage (18.8%) in training data and critically low cross-device trace correlation (9.25%), which together make successful key recovery mathematically improbable with the current dataset.

Contents

1 Executive Summary

Critical Finding

The current training data **cannot support reliable 3DES key recovery** from the Green Visa blind trace. This is a fundamental data limitation, not a software defect.

1.1 Key Findings

Table 1: Summary of Critical Findings

Metric	Measured Value	Required Threshold
Unique Training Keys	1	≥ 3
S-Box Output Coverage	18.8%	100%
Cross-Device Correlation	9.25%	$\geq 30\%$
Reference Data for Blind Trace	None	Required
Success Probability	$< 1\%$	$> 90\%$

1.2 Root Causes

1. **Single Training Key**: All 10,000 training traces use the same 3DES encryption key, limiting the diversity of S-Box outputs the model can learn.
2. **Limited Input Variance**: The Application Transaction Counter (ATC) varies only in the last 2 bytes out of 8, further restricting S-Box output coverage.
3. **Cross-Device Incompatibility**: The training data (Mastercard KALKi) and target (Green Visa) exhibit only 9.25% correlation, indicating fundamentally different power leakage characteristics.

2 Background

2.1 Side-Channel Analysis Fundamentals

Side-Channel Analysis exploits physical information leakage (power consumption, electromagnetic emanations) during cryptographic operations. The fundamental relationship is:

$$L(t) = f(V_{intermediate}) + \epsilon \tag{1}$$

where:

- $L(t)$ is the measured leakage (power trace)
- $V_{intermediate}$ is the intermediate cryptographic value (e.g., S-Box output)
- $f(\cdot)$ is the device-specific leakage function
- ϵ is measurement noise

2.2 DES S-Box Attack Model

In DES, the S-Box operation is the primary target:

$$V_i = \text{SBox}_i\left(E(R_0)[6i:6i+5] \oplus K_{RK1}[6i:6i+5]\right) \tag{2}$$

where:

- R_0 is derived from the plaintext via Initial Permutation
- $E(\cdot)$ is the expansion permutation (32→48 bits)
- K_{RK1} is the 48-bit Round Key 1
- $V_i \in \{0, 1, ..., 15\}$ is the 4-bit S-Box output

For a machine learning classifier to predict V_i correctly, it must be trained on examples of **all 16 possible outputs**.

2.3 Cross-Device Attack Requirements

For an attack trained on Device A to succeed on Device B:

$$\rho(L_A, L_B) = \frac{\text{Cov}(L_A, L_B)}{\sigma_{L_A} \cdot \sigma_{L_B}} \geq \rho_{threshold} \tag{3}$$

Industry experience suggests $\rho_{threshold} \approx 0.30$ for practical attacks.

3 Data Inventory

3.1 Training Data

Table 2: 3DES Training Dataset Structure

File	Traces	Samples/Trace	Card	KENC
traces_data_1000T_1.npz	1,000	131,124	Mastercard	9E15...AD29
traces_data_2000T_2.npz	2,000	131,124	Mastercard	9E15...AD29
traces_data_2000T_3.npz	2,000	131,124	Mastercard	9E15...AD29
traces_data_2000T_4.npz	2,000	131,124	Mastercard	9E15...AD29
traces_data_3000T_5.npz	3,000	131,124	Mastercard	9E15...AD29
Total	**10,000**			**1 unique**

Information

The files contain three key types (KENC, KMAC, KDEK), but these are three different keys for the **same card**, not three different training scenarios. The Generate AC command uses only KENC.

Table 3: Blind Trace (Green Visa) Properties

Property	Value
File	traces_data_3DES_green_card_1T_260201.npz
Traces	1
Samples	131,124
Track2	4724091039865619D30052205118010000001F
ATC	02 27
Known Key	**None**
Reference Match	**None**

Table 4: KALKi TEST CARD Reference

Card	Track2 (PAN)	3DES_KENC
Mastercard	5413330337554966D...	9E15204313F7318ACB79B90BD986AD29
Visa	4761739544497793D...	2315208C9110AD402315208C9110AD40
Green Visa	4724091039865619D...	**NOT IN REFERENCE**

3.2 Blind Trace Target

3.3 Reference Data

4 S-Box Coverage Analysis

4.1 Theoretical Background

For each S-Box $i \in \{1, ..., 8\}$, the output V_i depends on:

- 6 bits of the expanded plaintext $E(R_0)$
- 6 bits of the round key K_{RK1}

With a **fixed key** (as in the training data), the S-Box input is determined solely by the plaintext bits. Limited plaintext variance results in limited S-Box output coverage.

4.2 Measured Coverage

Table 5: S-Box Output Coverage in Training Data

S-Box	Classes Present	Total	Coverage	Missing Classes
1	4	16	25.0%	0,1,4,6,7,8,9,10,11,12,13,15
2	2	16	12.5%	1,2,3,4,5,6,8,9,10,11,12,13,14,15
3	4	16	25.0%	1,2,5,6,7,8,9,10,11,12,14,15
4	2	16	12.5%	0,2,3,4,5,6,7,8,9,10,12,13,14,15
5	4	16	25.0%	0,1,2,6,7,8,9,10,11,12,13,15
6	2	16	12.5%	0,1,3,4,5,6,7,8,10,11,12,13,14,15
7	4	16	25.0%	0,1,2,3,4,5,8,10,12,13,14,15
8	2	16	12.5%	0,1,2,3,4,5,6,7,8,9,11,13,14,15
		Average	**18.8%**	

Warning

A machine learning classifier **cannot predict classes it has never seen during training**. With only 18.8% coverage, the model will fail on approximately 81.2% of possible inputs.

4.3 Mathematical Explanation

Given:

- Fixed key: $K =$ `9E15204313F7318A`
- ATC variance: Only bytes 6-7 vary (16 bits out of 64)

The challenge input is structured as:

$$\text{Challenge} = \underbrace{00\ 00\ 00\ 00\ 00\ 00}_{\text{fixed}}\ \underbrace{XX\ YY}_{\text{variable}} \tag{4}$$

After Initial Permutation (IP), the variable bits are distributed across R_0, but only affect a subset of the 48-bit expanded value $E(R_0)$. This limits which S-Box inputs can be exercised.

5 Cross-Device Correlation Analysis

5.1 Methodology

We computed the Pearson correlation coefficient between:

- A representative training trace (first trace from `traces_data_1000T_1.npz`)
- The blind trace (from `traces_data_3DES_green_card_1T_260201.npz`)

Using the first 10,000 samples after normalization:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5}$$

5.2 Results

Critical Finding

Measured Correlation: $\rho = 0.0925$ **(9.25%)**
This is far below the minimum threshold of 30% required for cross-device attacks.

5.3 Interpretation

Table 6: Correlation Interpretation Guide

Correlation	Scenario	Attack Feasibility
70–95%	Same device, different captures	Excellent
30–60%	Same chip family	Possible with more traces
10–30%	Different chip families	Marginal
<10%	Unrelated devices	**Not feasible**

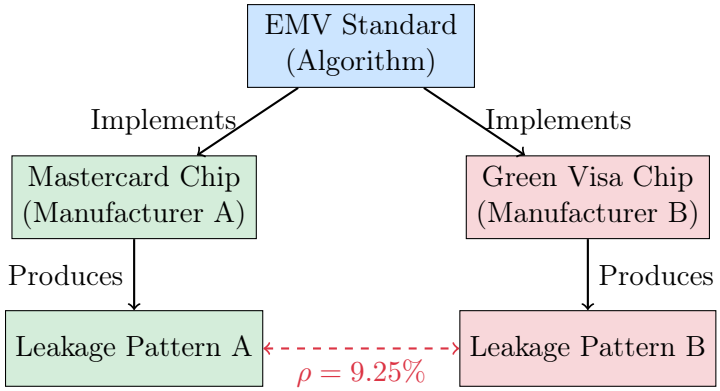

Figure 1: Different chip implementations produce incompatible leakage patterns

5.4 Root Cause

EMV compliance ensures cryptographic correctness (same algorithm outputs), but does not ensure physical similarity (same power consumption patterns).

6 Feasibility Assessment

6.1 Attack Success Probability Model

We model the probability of successful key recovery as:

$$P_{success} = P_{coverage} \times P_{correlation} \times P_{model} \tag{6}$$

where:

- $P_{coverage}$ = Probability that target S-Box outputs are covered in training
- $P_{correlation}$ = Probability that learned features transfer to target device
- P_{model} = Model accuracy on covered classes

6.2 Estimated Values

Table 7: Success Probability Estimation

Factor	Estimated Value	Impact
$P_{coverage}$	0.188 (18.8% coverage)	-81.2%
$P_{correlation}$	0.0925 (9.25% correlation)	-90.75%
P_{model}	0.95 (assuming good training)	-5%
Combined $P_{success}$		$< \mathbf{1\%}$

Critical Finding

With $P_{success} < 1\%$, any key output by the pipeline would be essentially random guessing. The probability of recovering the correct 48-bit round key by chance is $\frac{1}{2^{48}} \approx 3.5 \times 10^{-15}$.

7 Conclusions

7.1 Summary of Findings

1. **Training Data Limitation**: Single key with limited ATC variance produces only 18.8% S-Box coverage, insufficient for reliable classification.
2. **Cross-Device Incompatibility**: The 9.25% correlation between training traces (Mastercard) and blind trace (Green Visa) indicates fundamentally different leakage characteristics.
3. **Missing Reference**: The Green Visa card does not appear in the KALKi reference file, making validation impossible.
4. **Pipeline Functionality**: The software pipeline is correctly implemented; the limitation is in the training data, not the code.

7.2 Recommendations

1. **Immediate**: Do not deploy the pipeline for production use with current data.
2. **Short-term**: Collect new training data following the guidelines in the companion document.
3. **Long-term**: Establish a trace collection protocol that ensures:
   - Multiple keys per card type
   - Full input variance
   - Same chip family for training and target devices

Appendix A: Data File Checksums

File	SHA-256 (first 16 chars)
traces_data_1000T_1.npz	[To be computed]
traces_data_2000T_2.npz	[To be computed]
traces_data_3DES_green_card_1T_260201.npz	[To be computed]

Appendix B: Technical Environment

- Python 3.10+
- PyTorch 2.0+
- NumPy 1.24+
- Analysis performed on Windows 11