Practical Improvements of Profiled Side-Channel Attacks on a Hardware Crypto-Accelerator

M. Abdelaziz EL AABID Sylvain GUILLEY
< elaabid@telecom-ParisTech.fr >

TELECOM-ParisTech
Université Paris 8

AFRICACRYPT 2010 - Stellenbosch, South Africa

Outline

Introduction to side channel attacks

Cryptographic device + Cryptographic algorithm

$\Downarrow$

Adversary + Information leakage

$\Downarrow$

Side Channel attack

- DPA [KocherJaffeJun99],
- Template Attacks [ChariRaoRohatgi02],
- CPA [BrierClavierOlivier04],
- MIA [GierlichsBatinaTuylsPreneel08],

Security point of view

A chase between attacks and countermeasures:

- attacks more advanced
- what methodology for countermeasures: local or general, algorithm or implementation

Which theory?

- 2003: *'Physically Observable Cryptography'*, a first theoretical framework for SCA, by Silvio Micali and Leonid Reyzin
- 2009: *'A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks'*, F-X Standeart et al.
  - Information theory: the amount of information given by a leakage function
  - Adversary success: guessing entropy, success rate

A new protagonist: Security evaluator

The side channel evaluator defines two metrics:

1. robustness of the circuit and
2. strength of the adversary.

- adversary # evaluator: treated fairly?
- This element will be the basis of our experiments.

Prerequisites

Leakage Function $\mathcal{L}(C, M, R)$

is a function of three parameters:

- C is the current internal configuration of the circuit / algorithm;
- M is a set of measures;
- R is a random string which represents the noise.

divide-and-conquer strategy

to retrieve separately parts of the secret key.
$f : K \mapsto S_K$ which maps each key k onto a class $s_k = f(k)$ such that $|s_K| \ll |k|$.
In DES: $|s_K| = 6 \ll |k| = 56$.

Security and Robustness Metrics

Conditional entropy

The conditional entropy $\mathbf{H}(S_K \mid L)$ is defined by:

$$\mathbf{H}(S_K \mid L) \quad = \quad -\sum_{s_K} \Pr(s_K) \sum_{l} \Pr(l \mid s_K) \cdot \log_2 \Pr(s_K \mid l)\,. \quad (1)$$

We define the conditional entropy matrix:

$$\mathbf{H}_{s_K, s_{Kc}} \doteq -\sum_{l} \Pr(l \mid s_K) \cdot \log_2 \Pr(s_{Kc} \mid l)\,, \quad (2)$$

where s_K and s_{Kc} are respectively the good subkey and the subkey candidate.

Metrics

Conditional entropy

from (1) and (2):

$$\mathbf{H}(S_K \mid L) = \sum_{s_K} \Pr(s_K)\ \mathbf{H}_{s_K, s_K} \tag{3}$$

Success rate

An adversary is an algorithm that aims at guessing a key class s_K with high probability. The success rate is estimated from the number of times the attack is successful.

Template attack

Phase 1: Profiling

- adversary chooses a sensitive variable
- experiments all values $V_i, i \in [0, N]$ of this variable
- collects L traces $\forall V_i, i \in [0, N]$ in some sets $\mathcal{S}_i, i \in [0, N]$
- computes the averages μ_i and covariance matrices Σ_i $\forall V_i, i \in [0, N]$
  - $$\mu_i = \frac{1}{|\mathcal{S}_i|} \sum_{t \in \mathcal{S}_i} t \quad \text{and} \quad \Sigma_i = \frac{1}{|\mathcal{S}_i| - 1} \sum_{t \in \mathcal{S}_i} (t - \mu_i)(t - \mu_i)^{\mathsf{T}} \quad (4)$$
  - (μ_i, Σ_i) is called the template associated with value k of the subkey

Huge dataset

- requires a lot of memory
- calculation errors
- covariance matrices badely conditioned

Principal Component Analysis (PCA)

by a linear transformation of variables (change of coordinates) we can:

- Simplify the structure of correlation
- Reduce the size of the dataset

Template attack in PCA

In practice, only a few eigenvectors are sufficient to represent all data samples.
Let V be the matrix containing the most significant eigenvectors.

averages in new basis

$$\nu_k = V^{\mathsf{T}} \mu_k$$

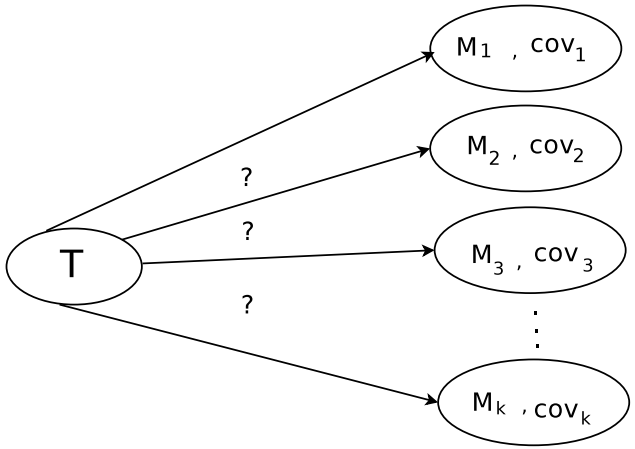
covariance matrices in new basis

$$\Lambda_k = V^{\mathsf{T}} \Sigma_k V$$

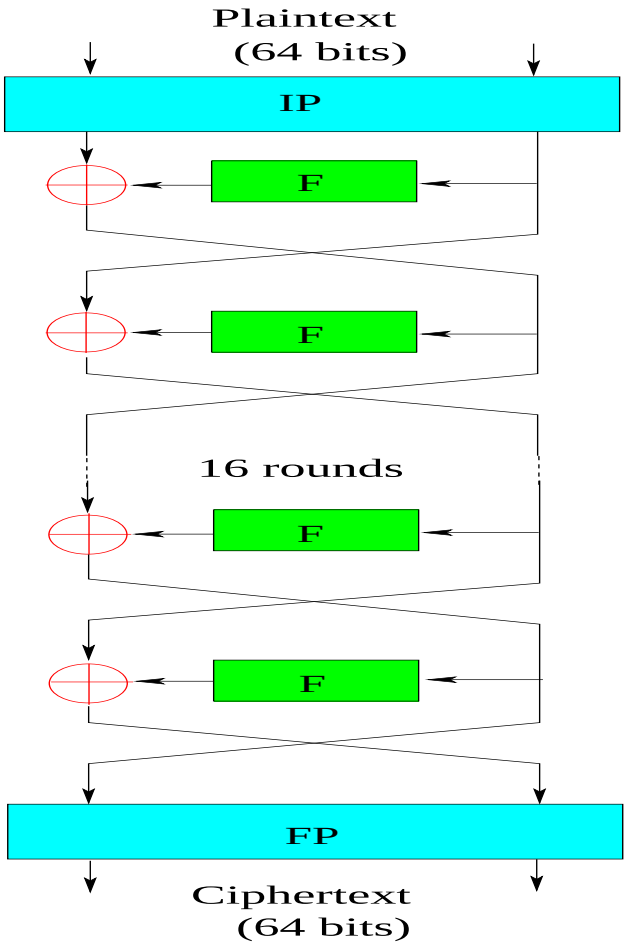
Template attack in PCA

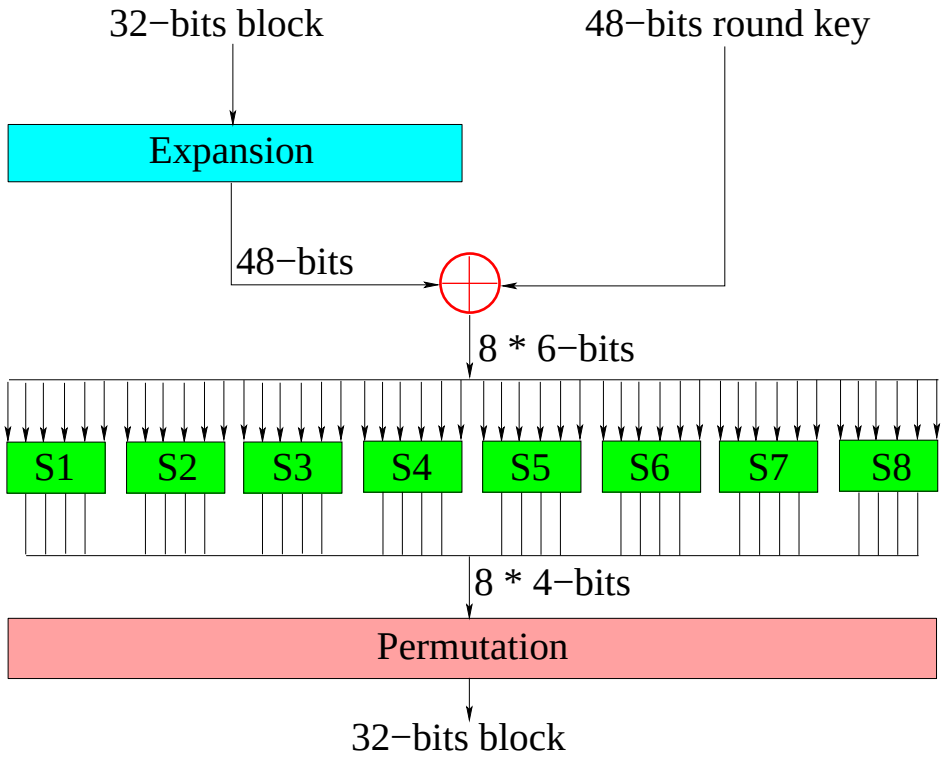
Phase 2: The online attack

- use Maximum Likelihood to select the most appropriate template for the trace T
- project T in the new database $\Rightarrow T_p = V^{\mathsf{T}} \times T$
- for each template (ν_i, Λ_i) calculate the probability density of T_p:
  - $p(T_p) = \frac{1}{\sqrt{(2\pi)^N |\Lambda_i|}} \exp\left(-\frac{1}{2}(T_p - \nu_i)^{\mathsf{T}} \Lambda_i^{-1} (T_p - \nu i)\right)$

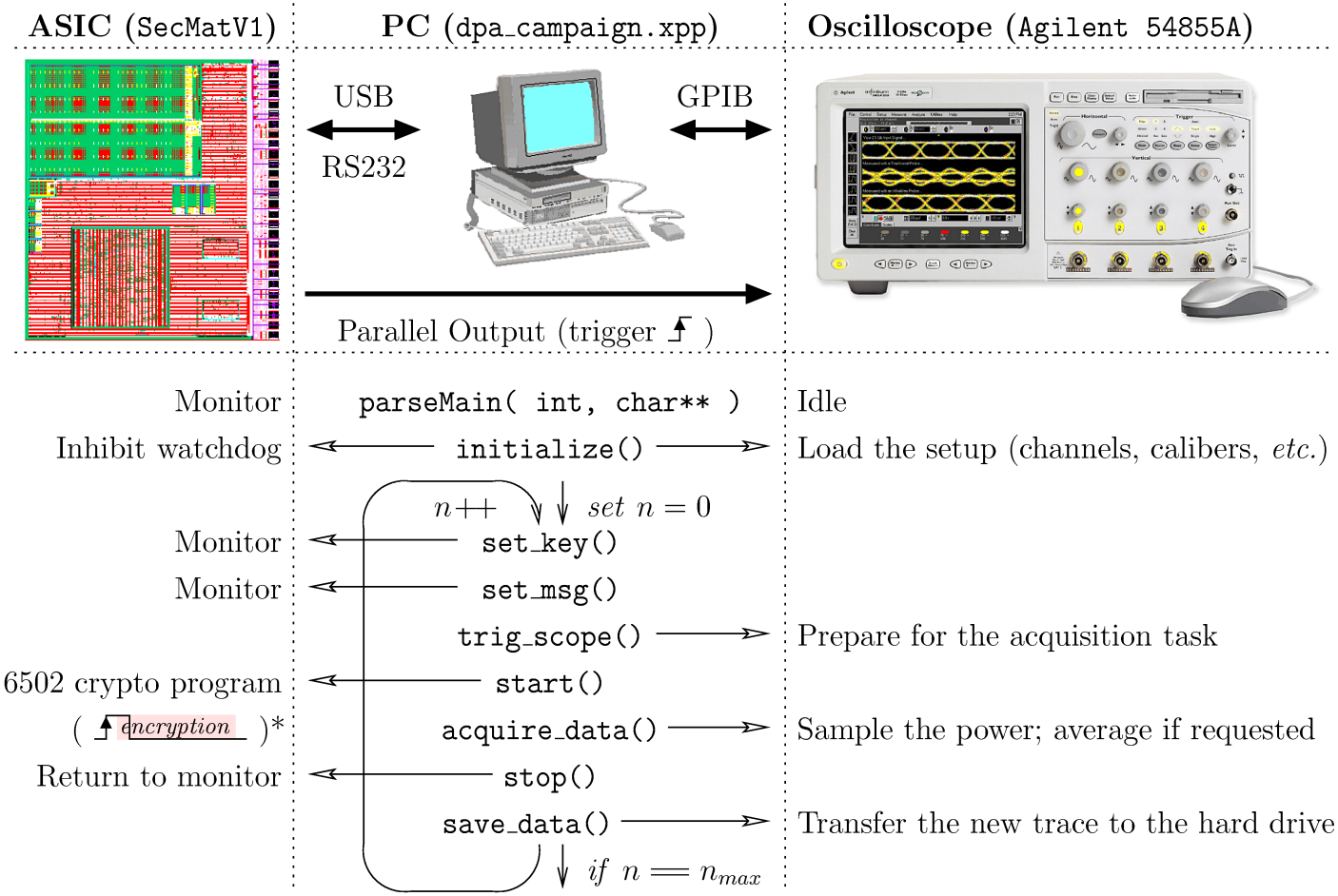


Data encryption standard (DES)

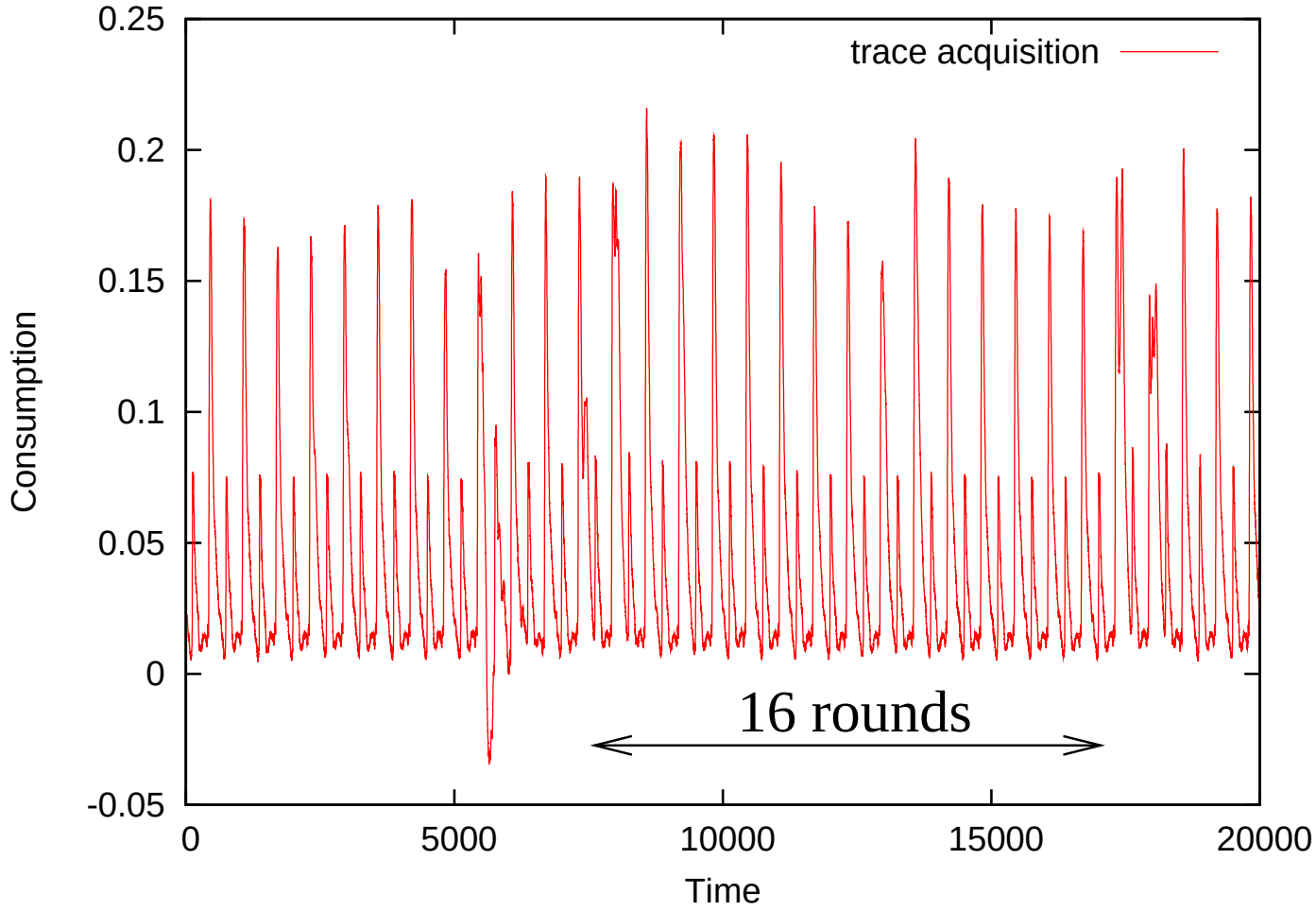




Acquisition



Example trace acquisition

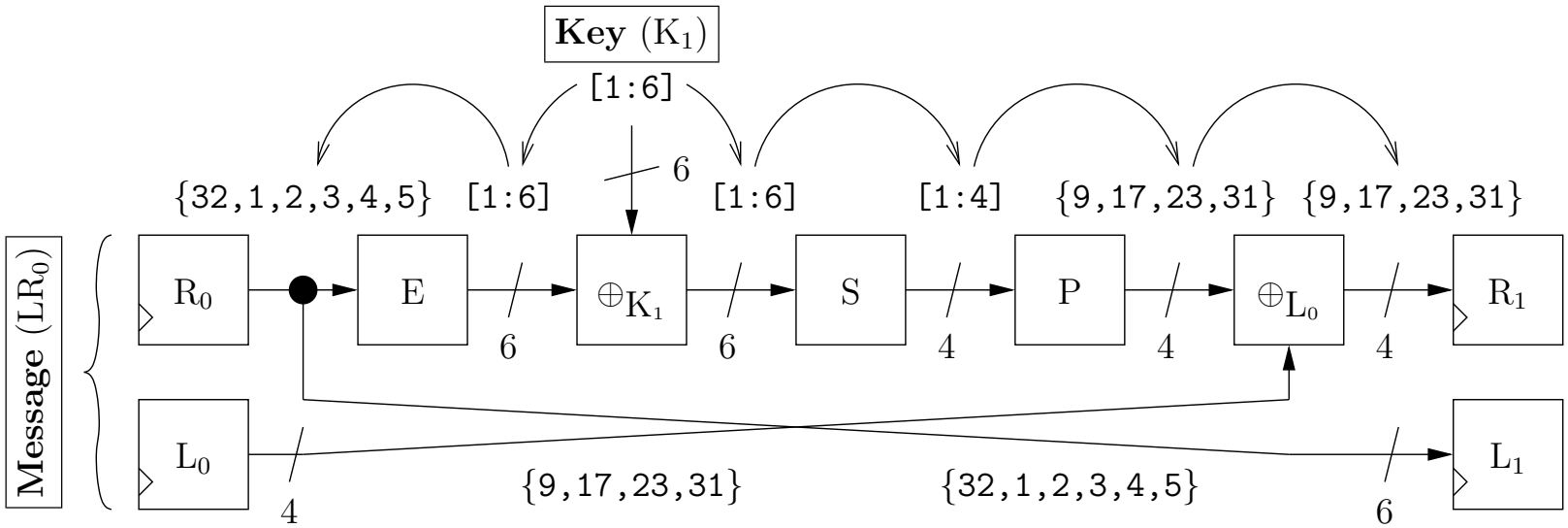


Which sensitive variable to choose?

- Algorithm dependency
  - input or output of an sbox?
- Implementation dependency
  - plain values or distance between two consecutive values?
  - hamming distance? plain distance?

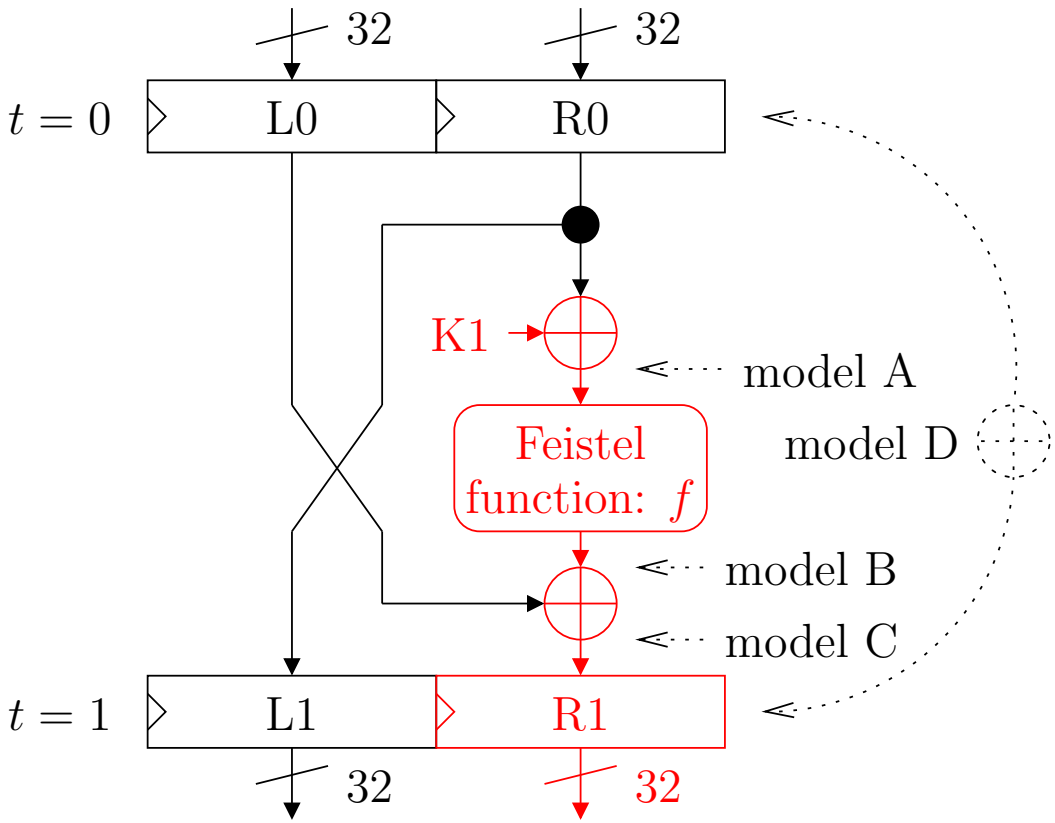
Leakage models

- Model A: the input of the first sbox, a 6-bit model.
- Model B: the output of the first sbox, a 4-bit model.
- Model C: the value of the first round (the fanout of the first sbox), a 4-bit model.
- Model D: the transition of model C.
- Model E: the Hamming weight of the model D.



Various leakage models for DES (iterative architecture)

Attack on the first round of DES



Caption: black = known values; red = unknown sensitive values

Eigenvectors

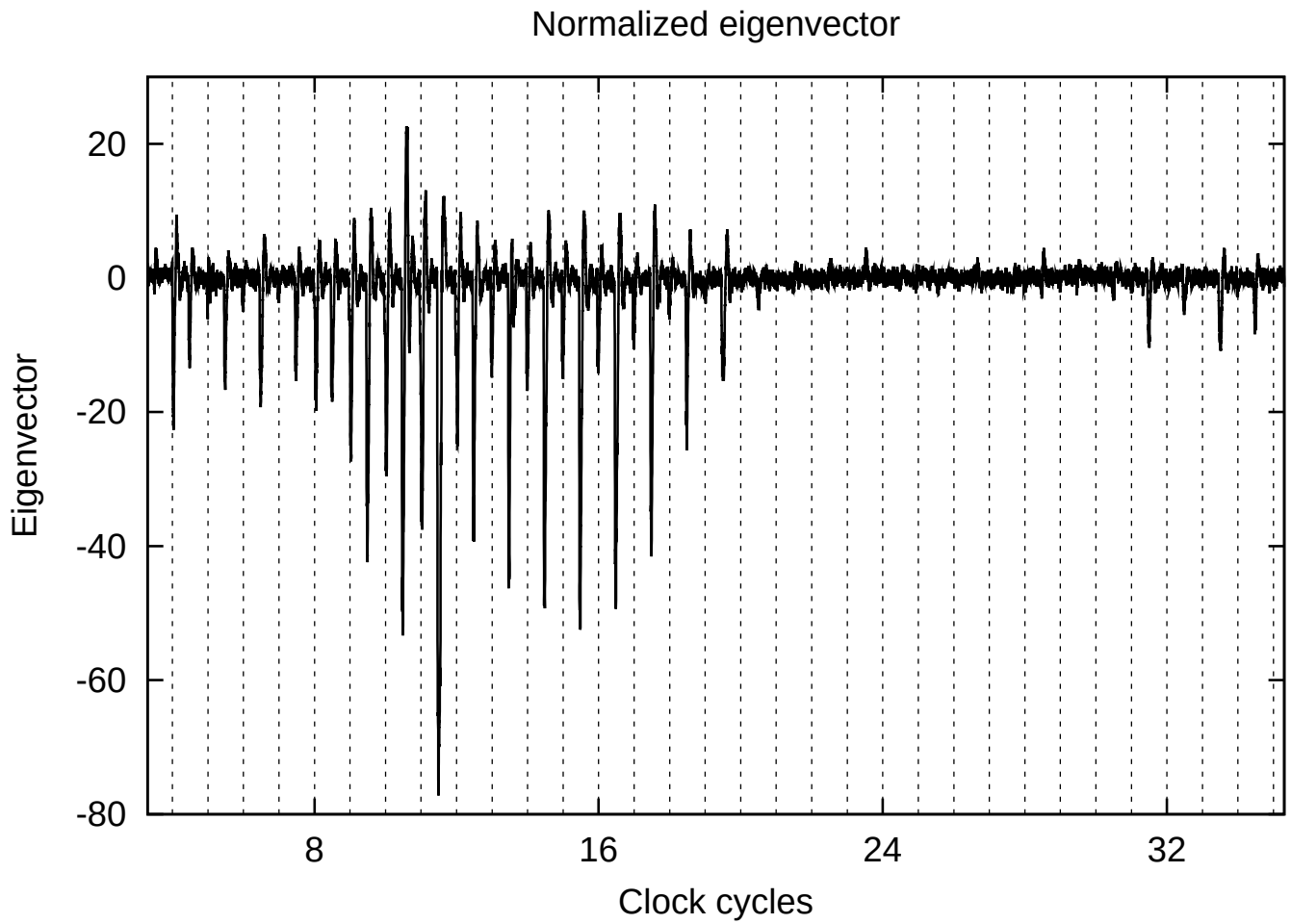


Figure 1: eigenvector for model A

Eigenvectors

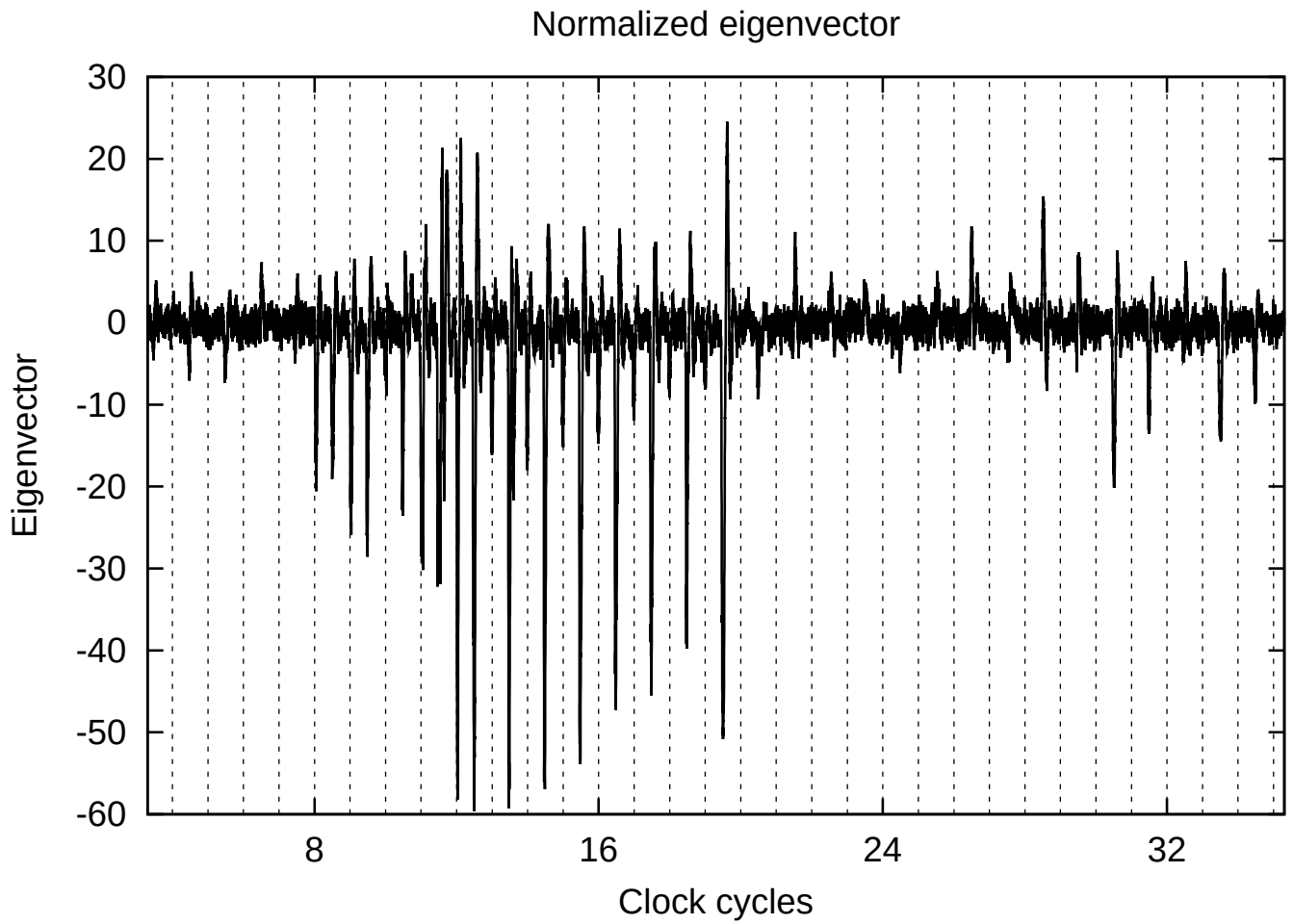


Figure 2: eigenvector for model B

Eigenvectors

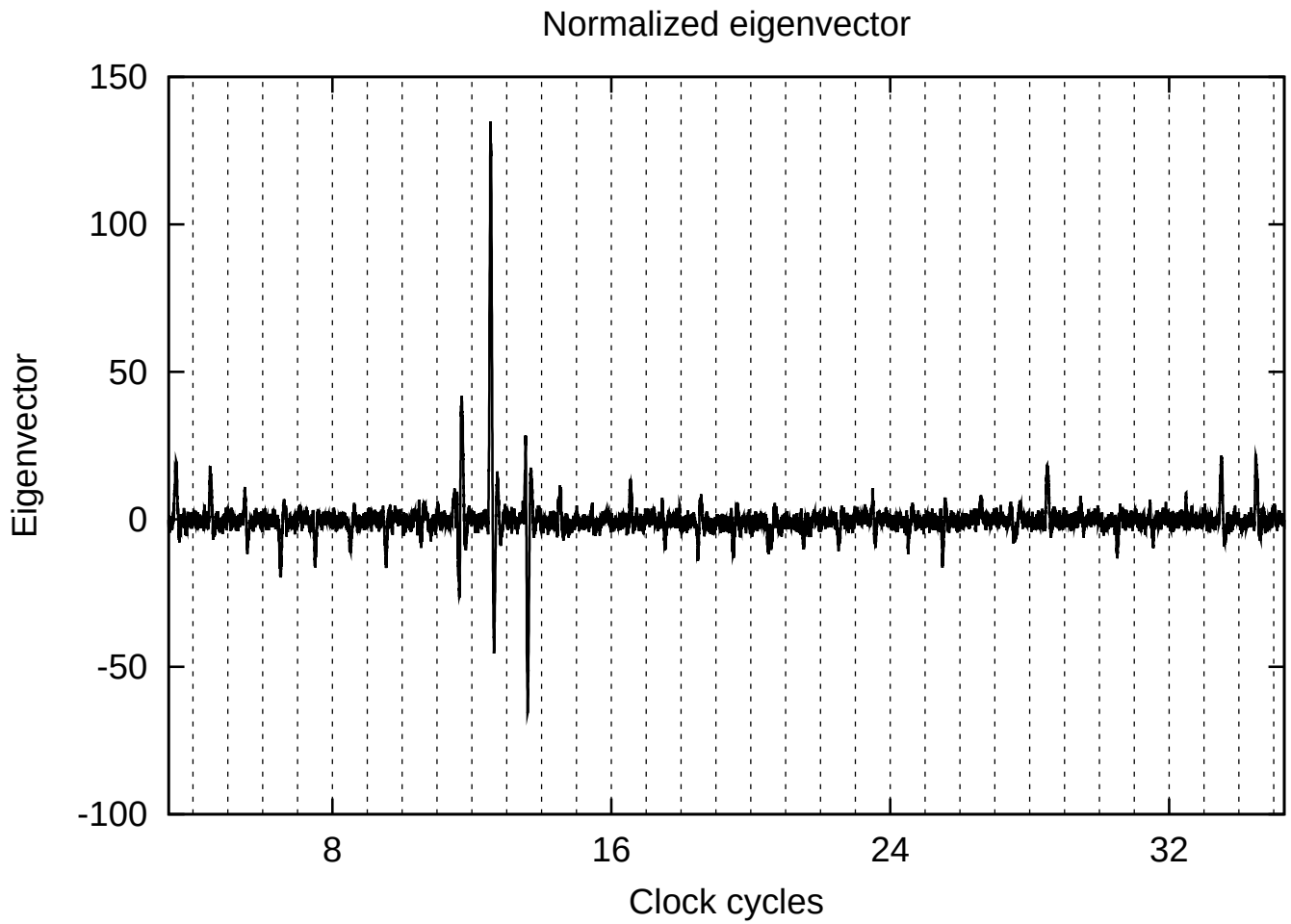


Figure 3: eigenvector for model C

Eigenvectors

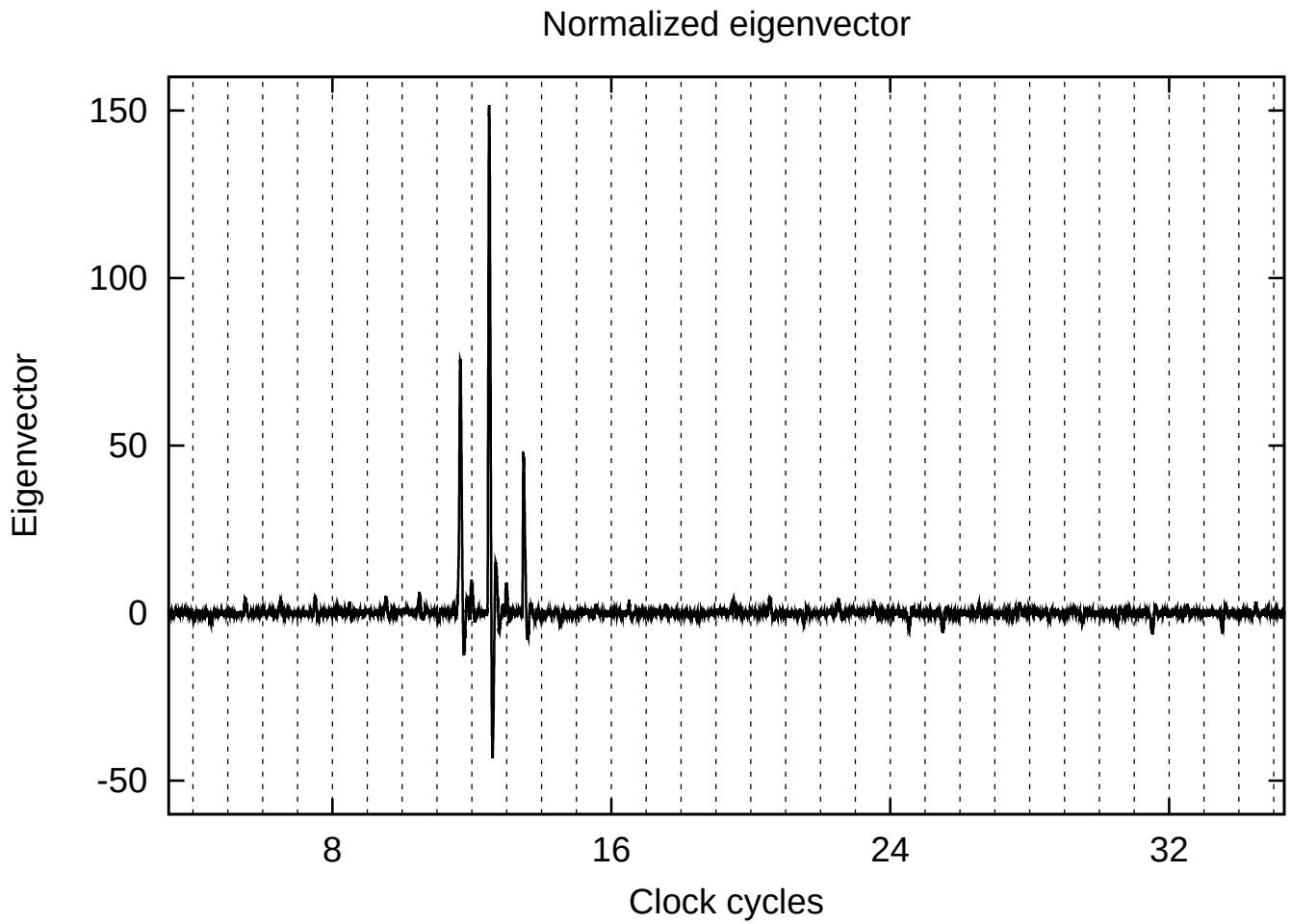


Figure 4: eigenvector for model D

Eigenvectors

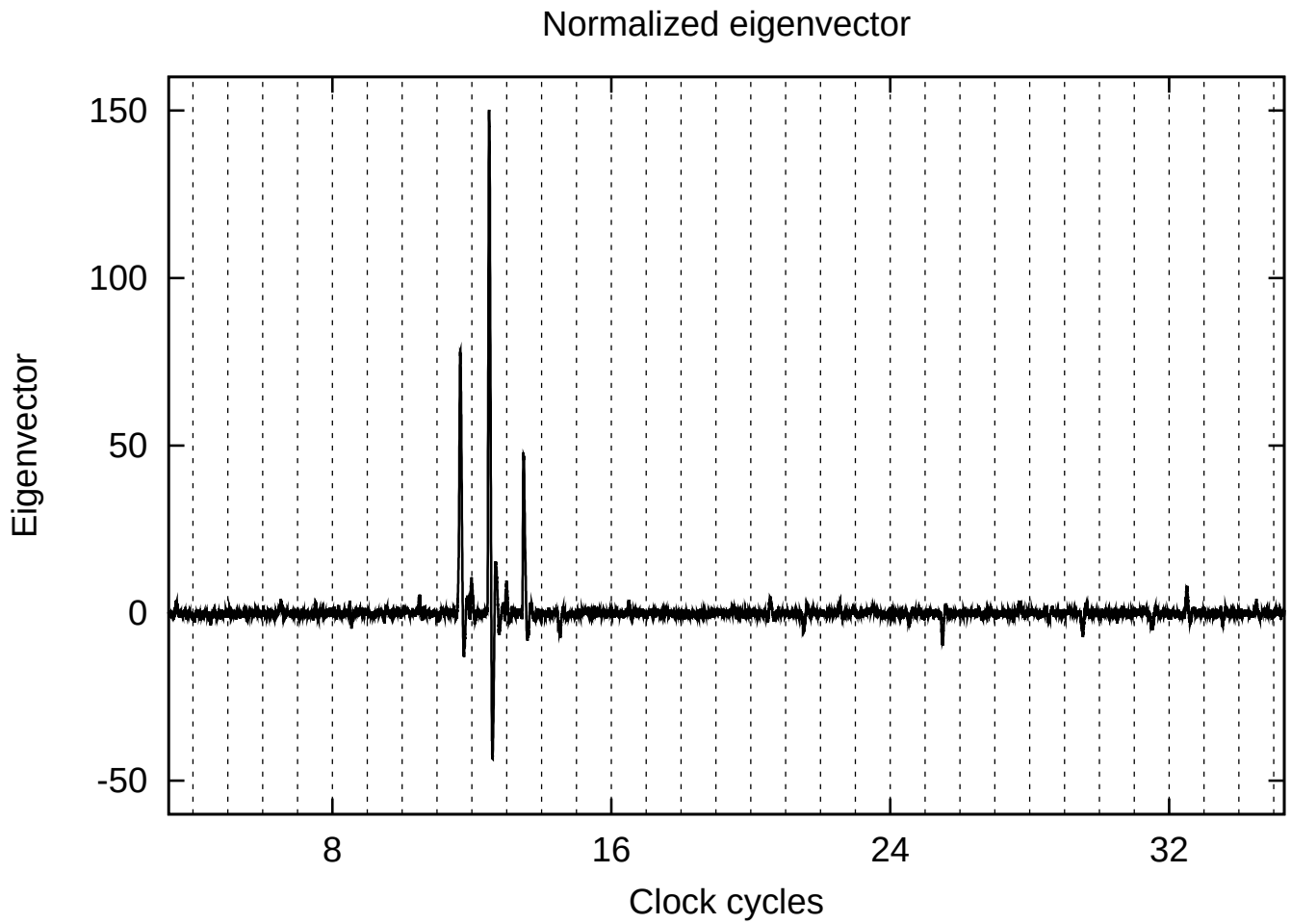


Figure 5: eigenvectors for model E

Success rate

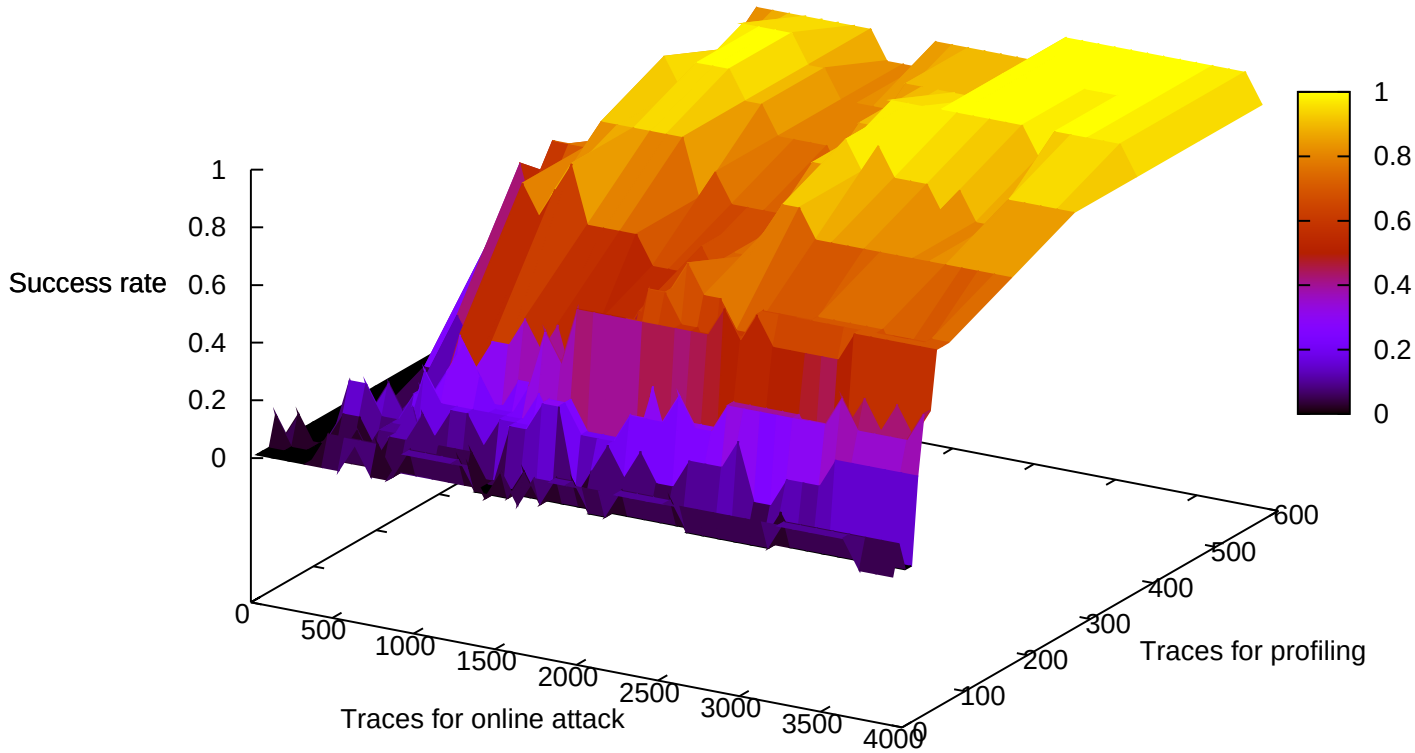


Figure 6: Success rate for model A

Success rate

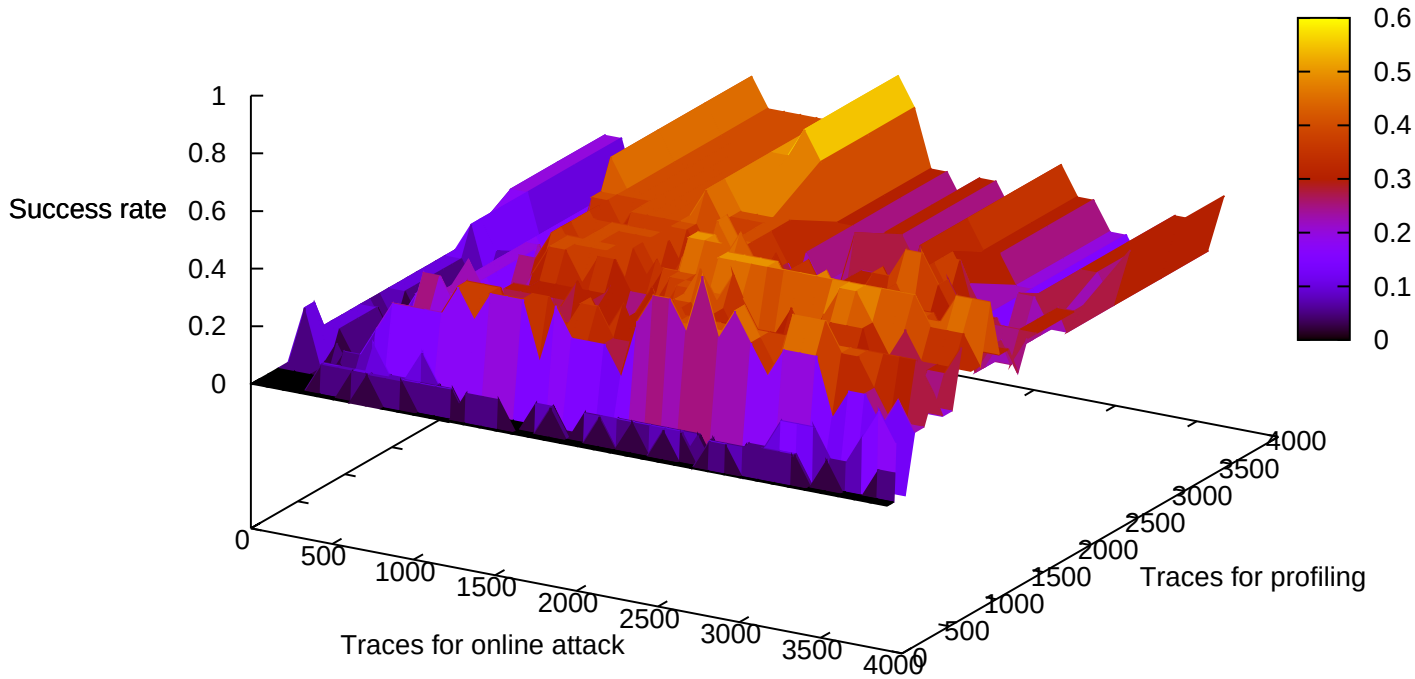


Figure 7: Success rate for model B

Success rate

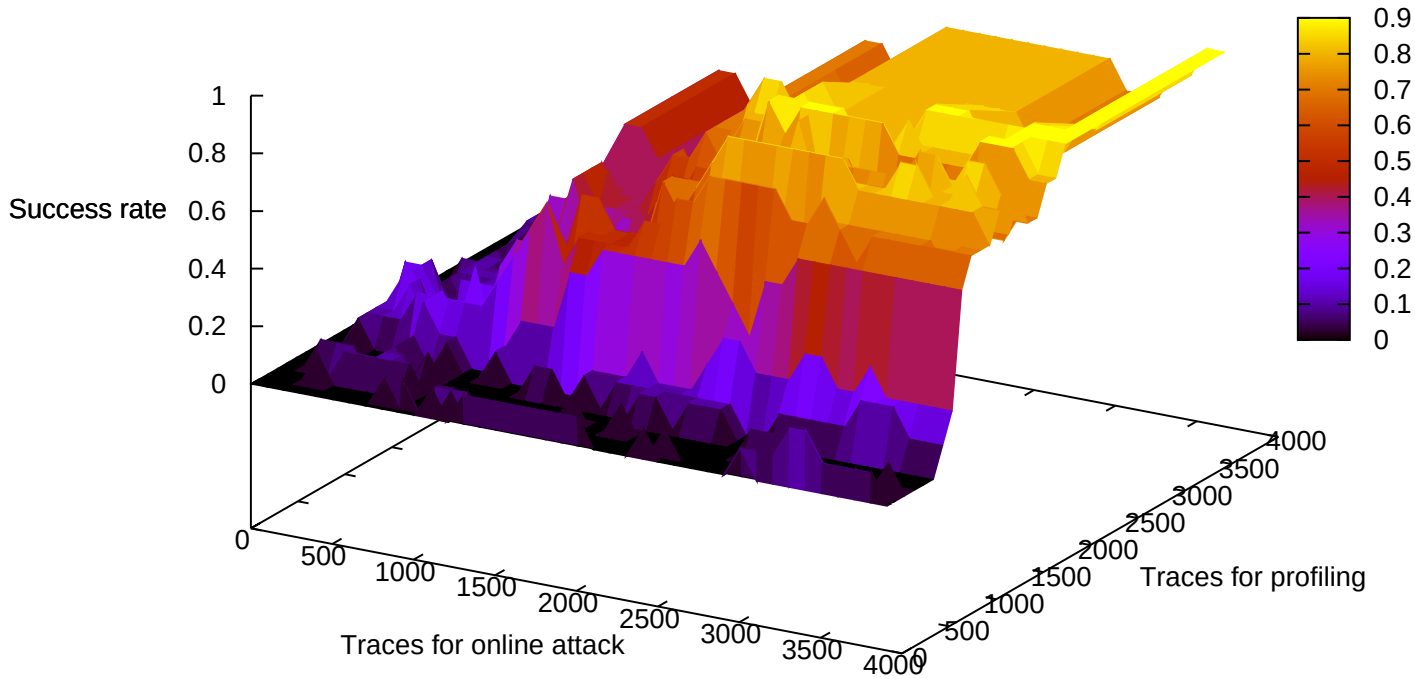


Figure 8: Success rate for model C

Success rate

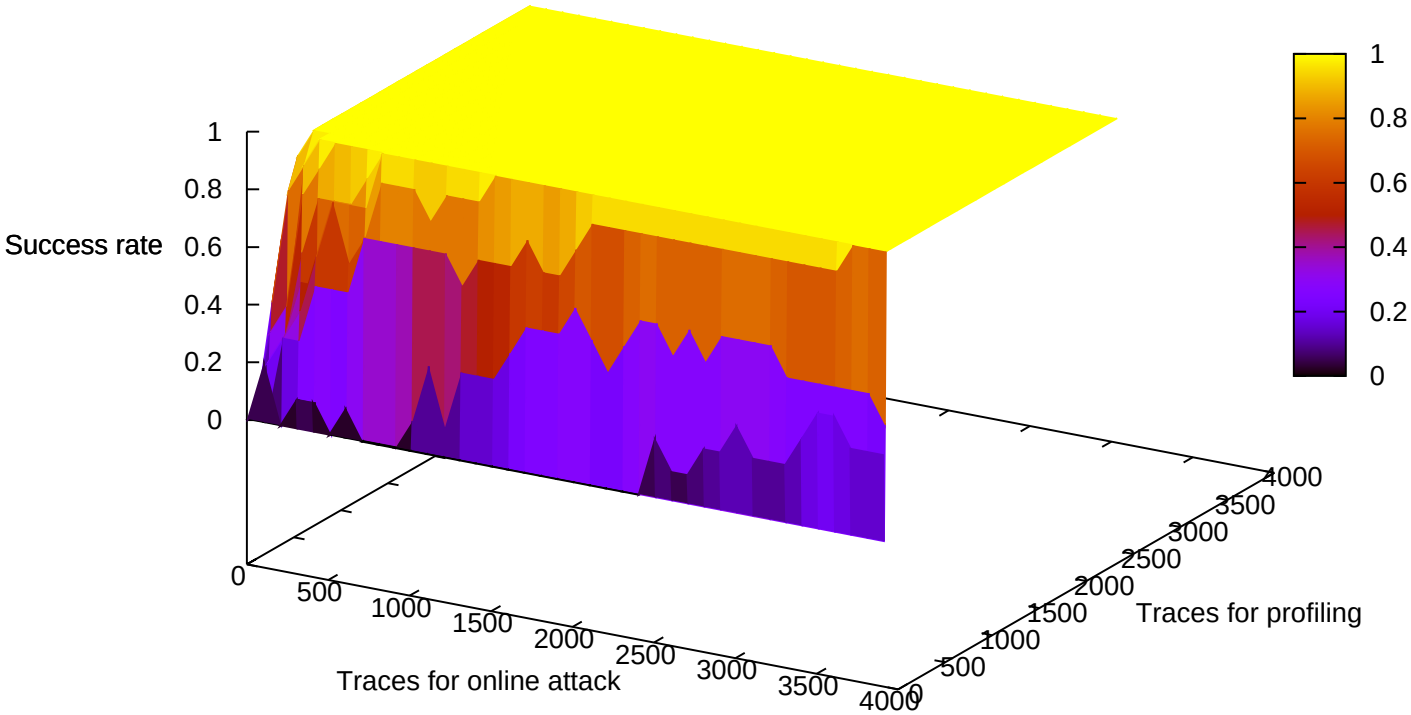


Figure 9: Success rate for model D

Success rate

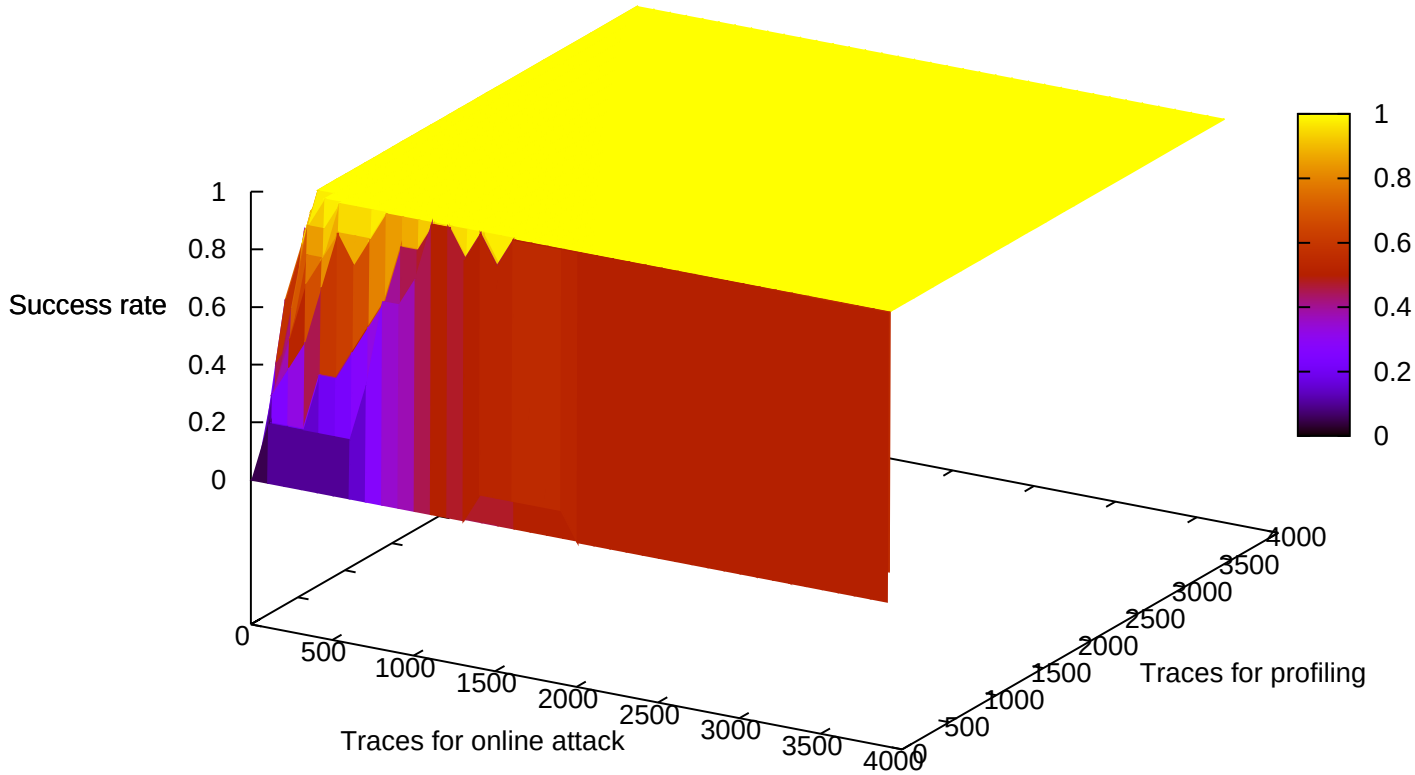


Figure 10: Success rate for model E

Metrics Comparison

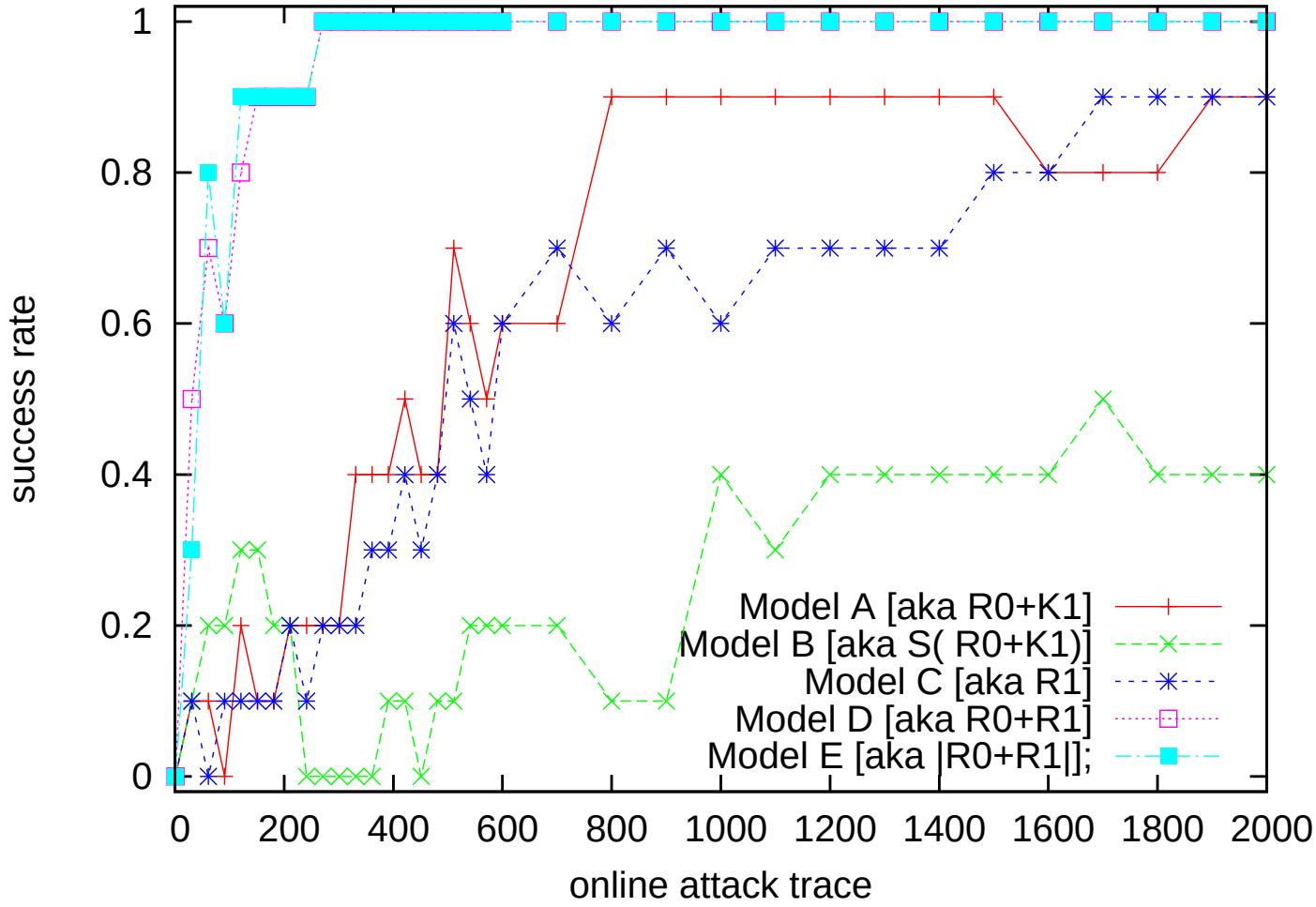


Figure 11: Success rate comparison

Metrics comparison

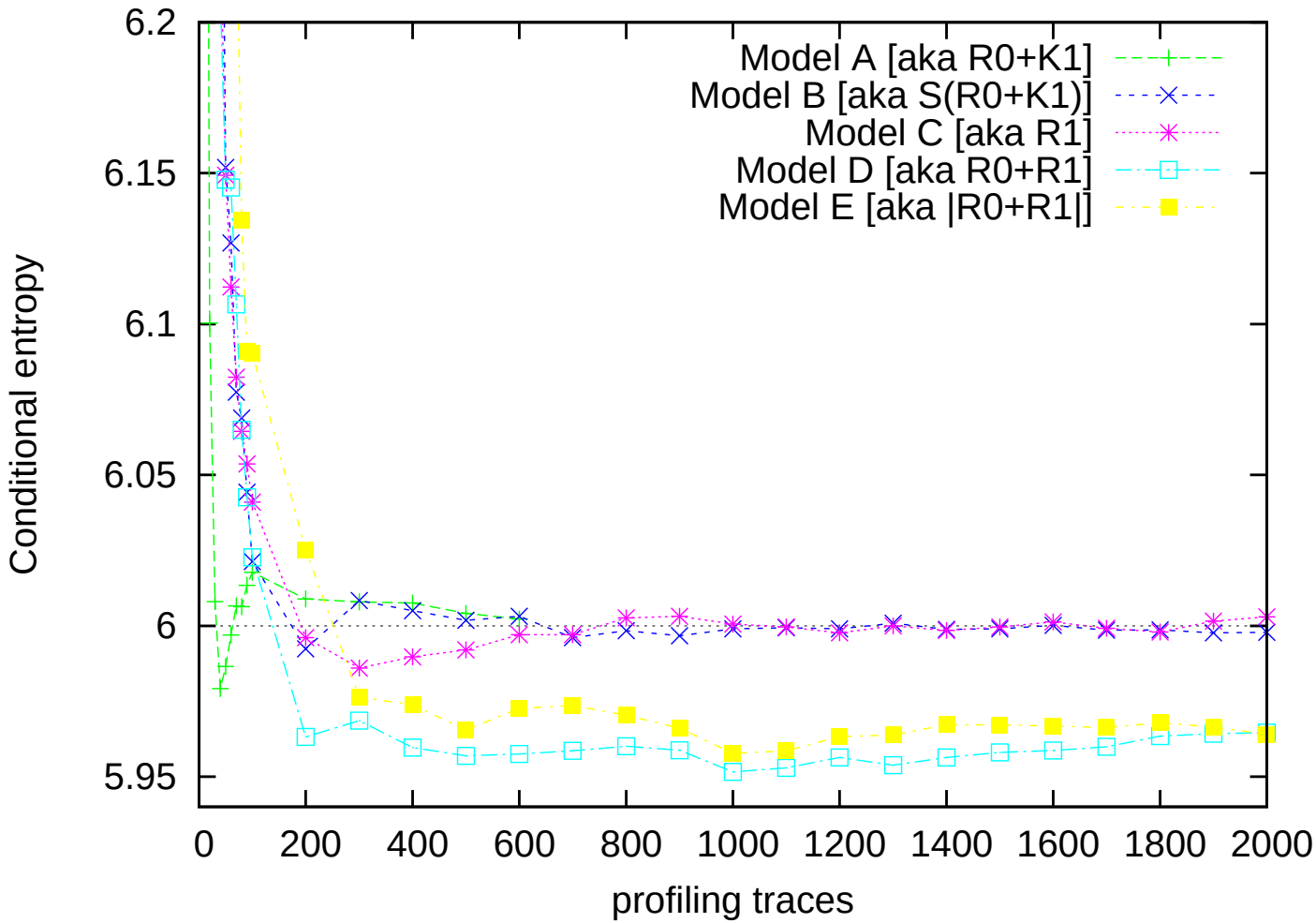
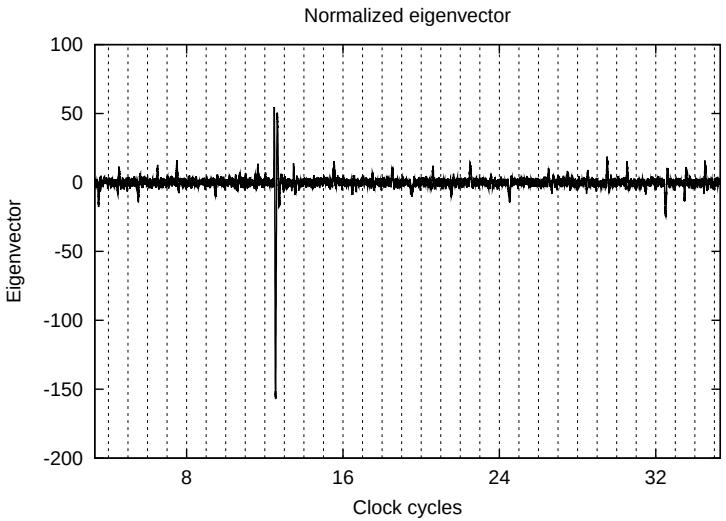
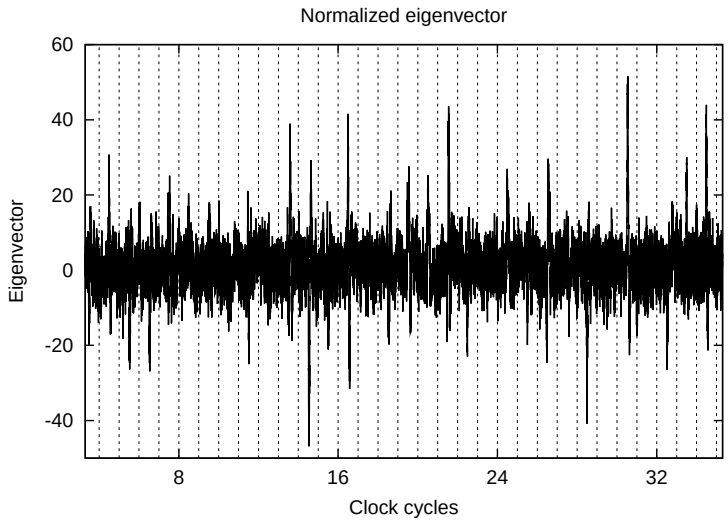


Figure 12: Conditional entropy comparison

Attack improvements



(a) Good eigenvector (1st).



(b) Bad eigenvector (12th).

Figure 13: Difference between 'good' and 'bad' eigenvector for model B.

Thresholding

- Eliminate noise
- Keep the moments of interest
- Reduce traces for profiling

Algorithm 1 thresholding

Require: $V = (v_0, v_1, .., v_i, .., v_n)$ are the eigenvectors

$th \in [0, 1]$ is the threshold

$max = \max_{0 \leq i \leq n} v_i$

for $i = 0$ to n **do**

  if $v_i < max * th$ **then**

    $v_i = 0$

  end if

end for

return V

Thresholds

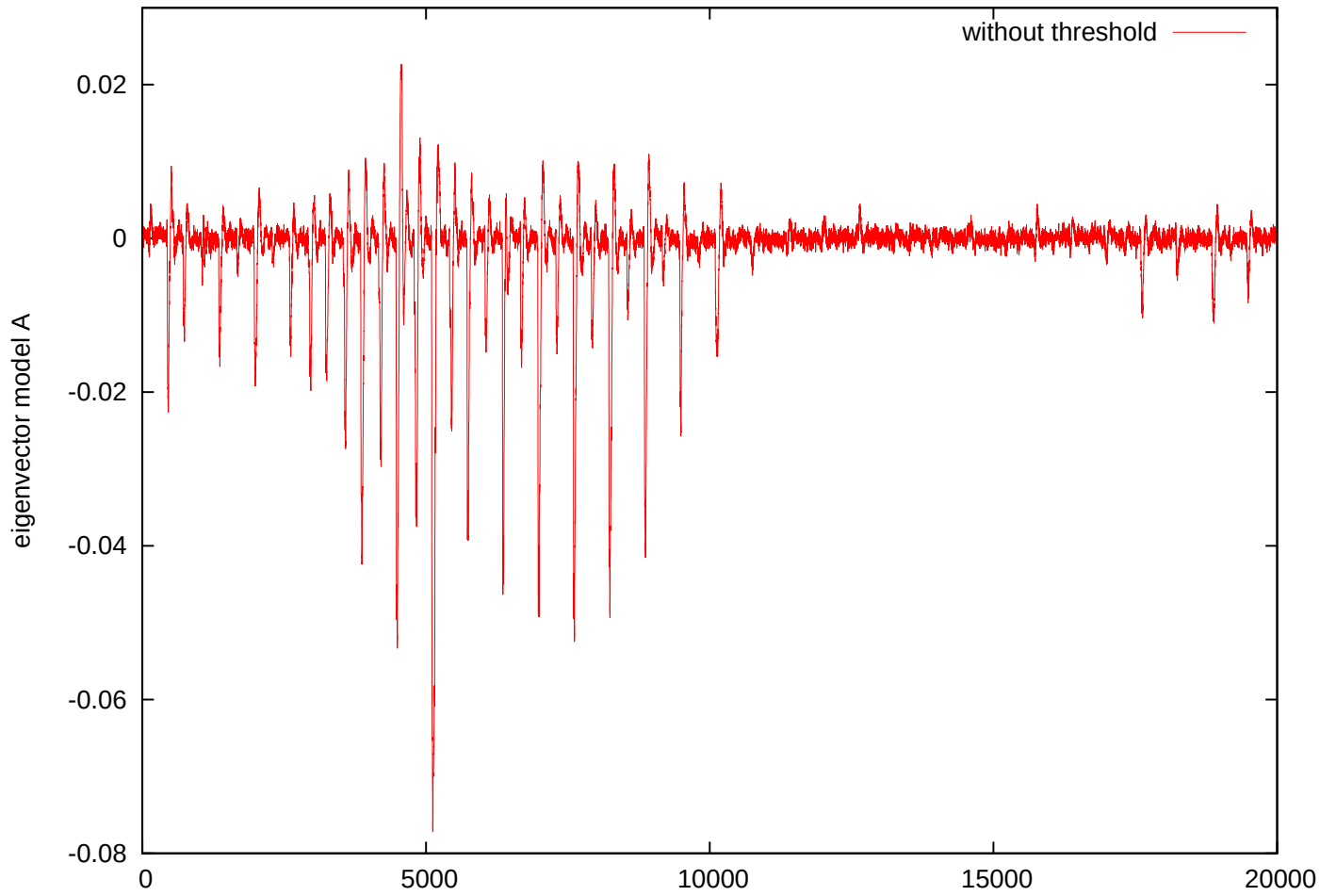


Figure 14: What is the best threshold?

Thresholds

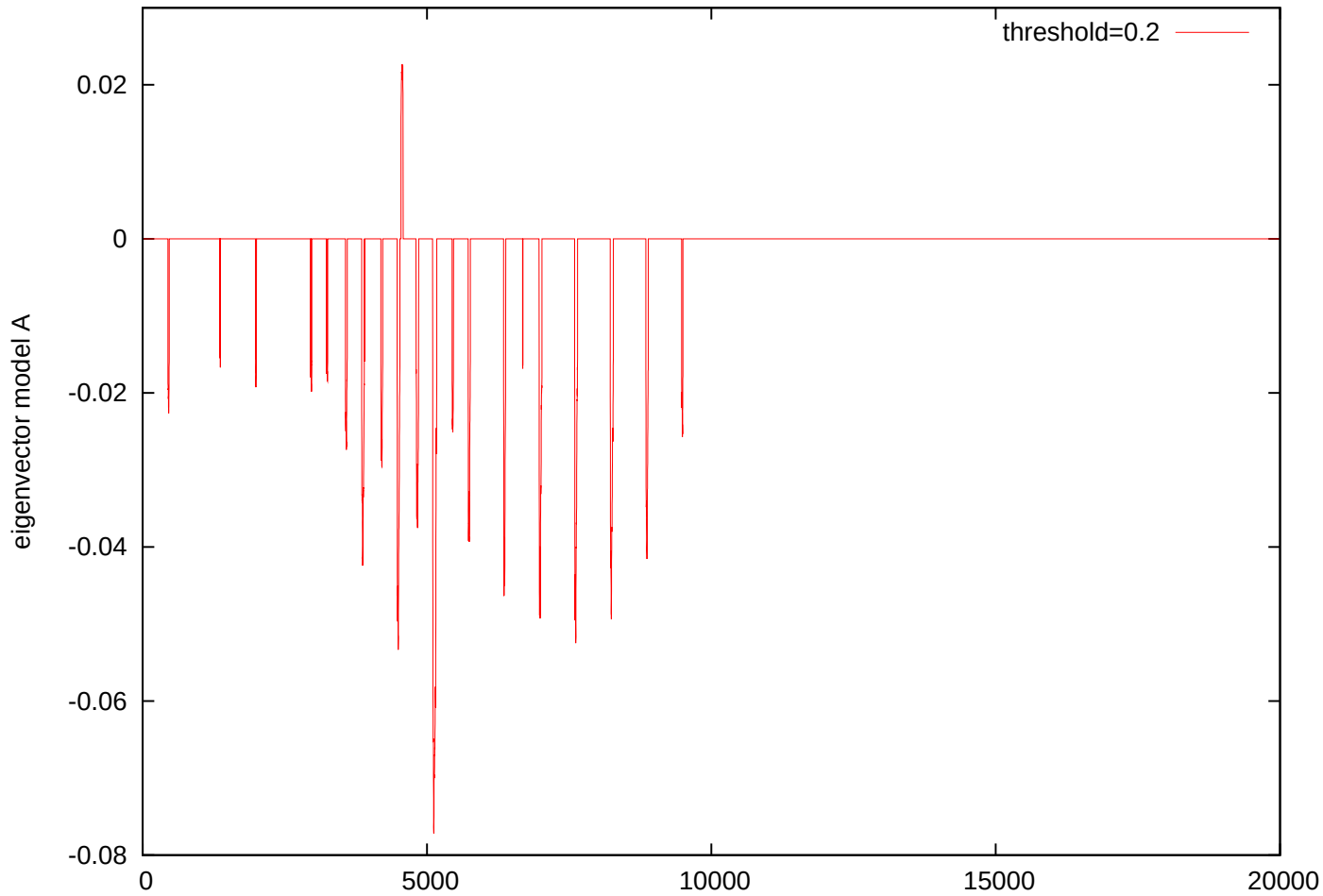


Figure 15: What is the best threshold?

Thresholds

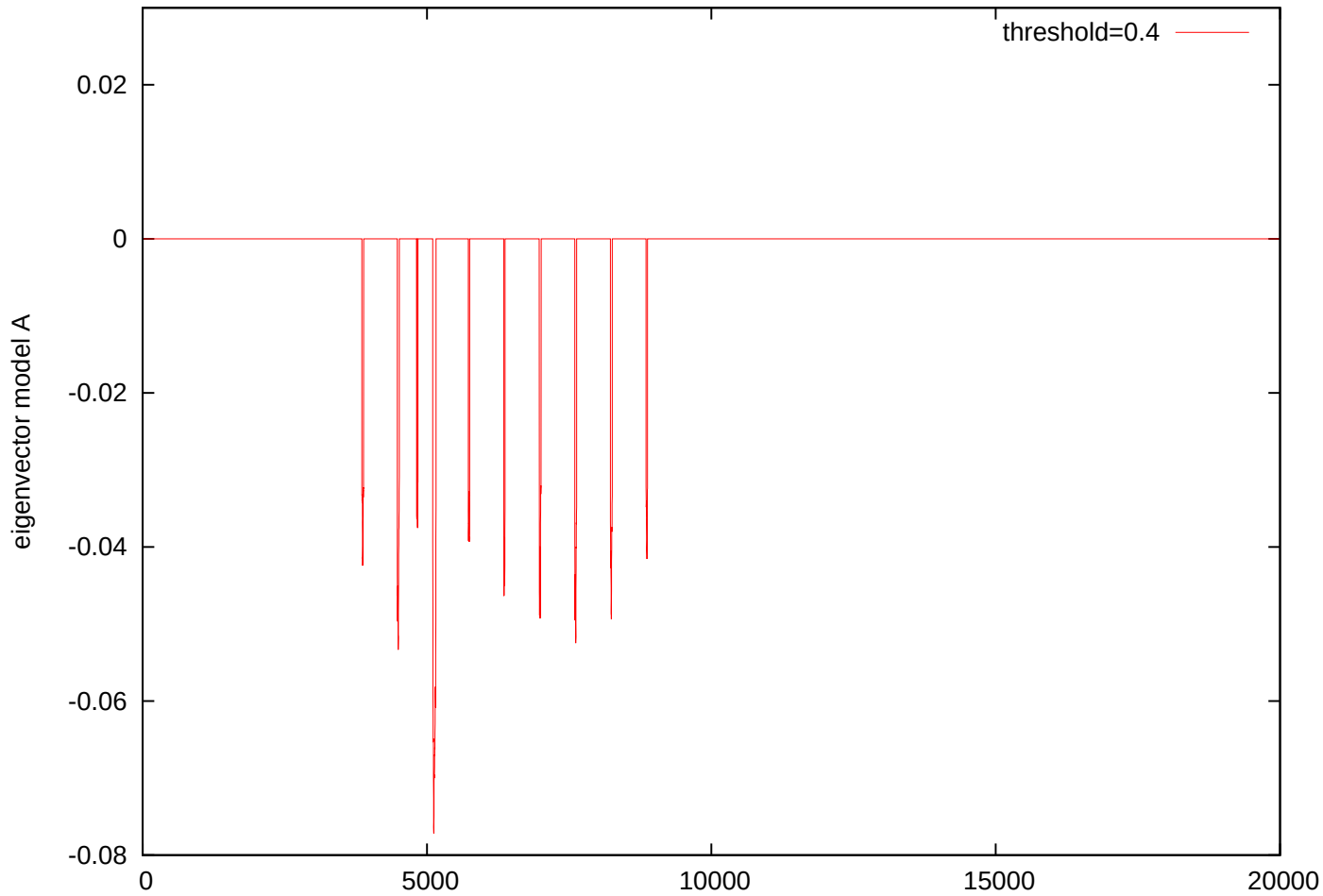


Figure 16: What is the best threshold?

Thresholds

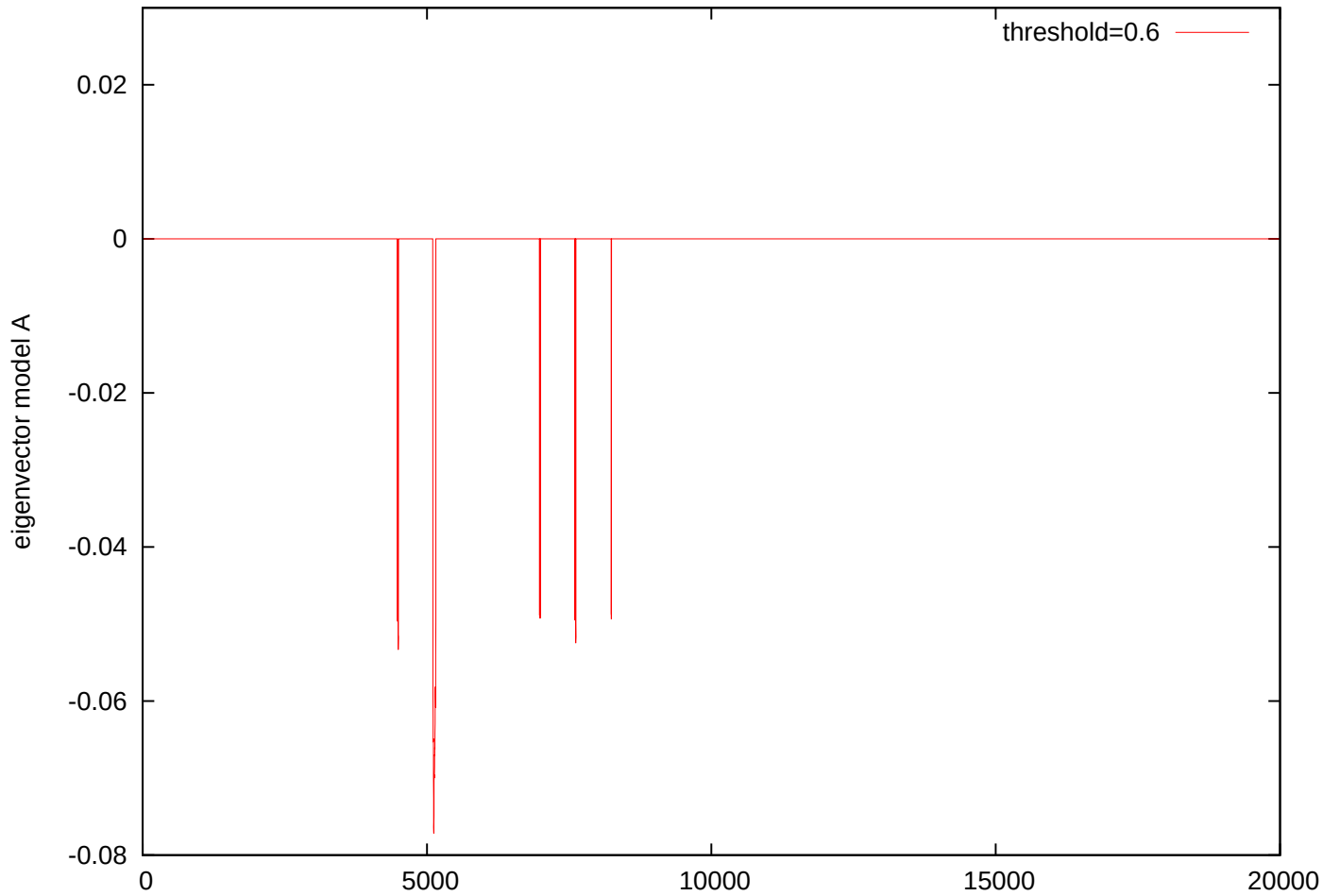


Figure 17: What is the best threshold?

Thresholds

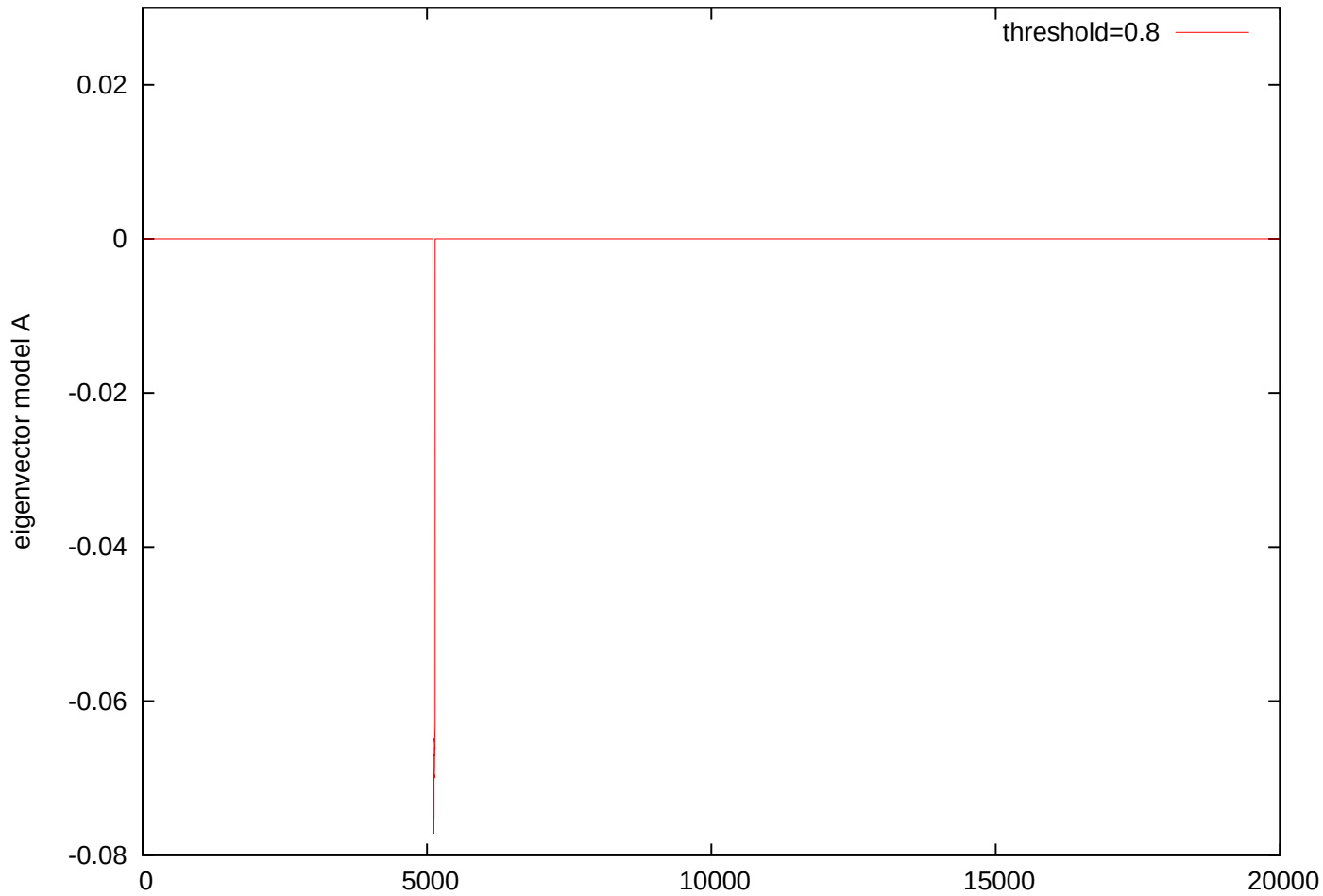
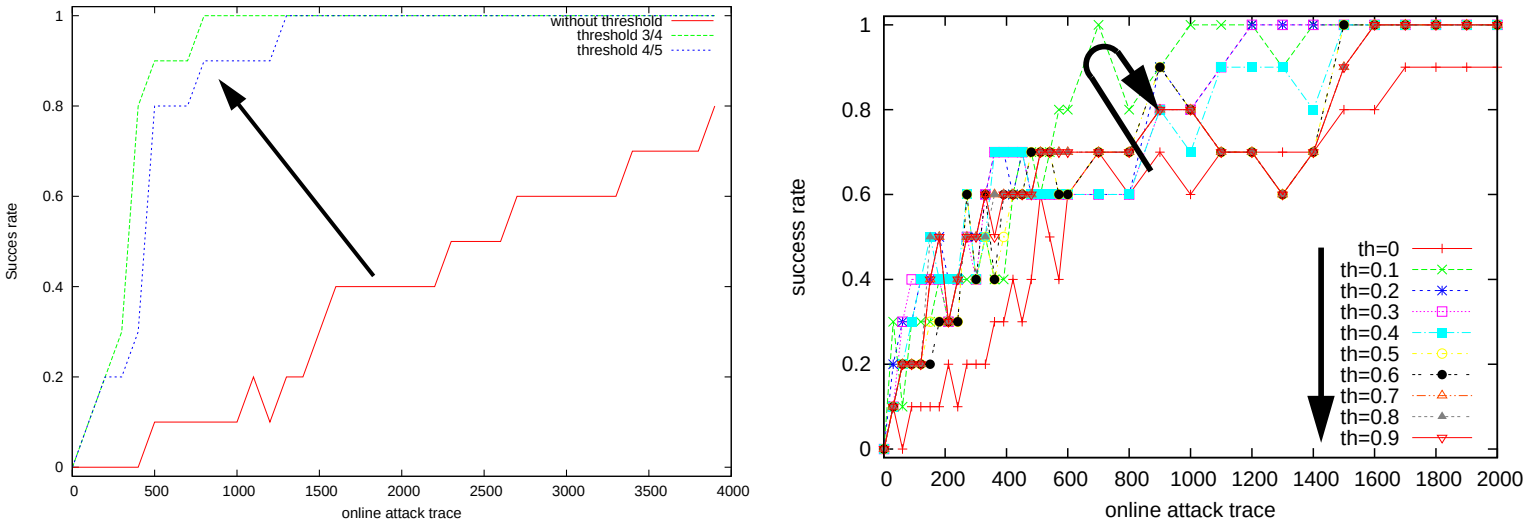


Figure 18: What is the best threshold?



(a) (b)

Figure 19: Success rate comparison (a) without and with threshold for model A and (b) with different thresholds for model C.

Evaluator vs Adversary

- fair conditions: same model
- adversary advantage: new attack trick

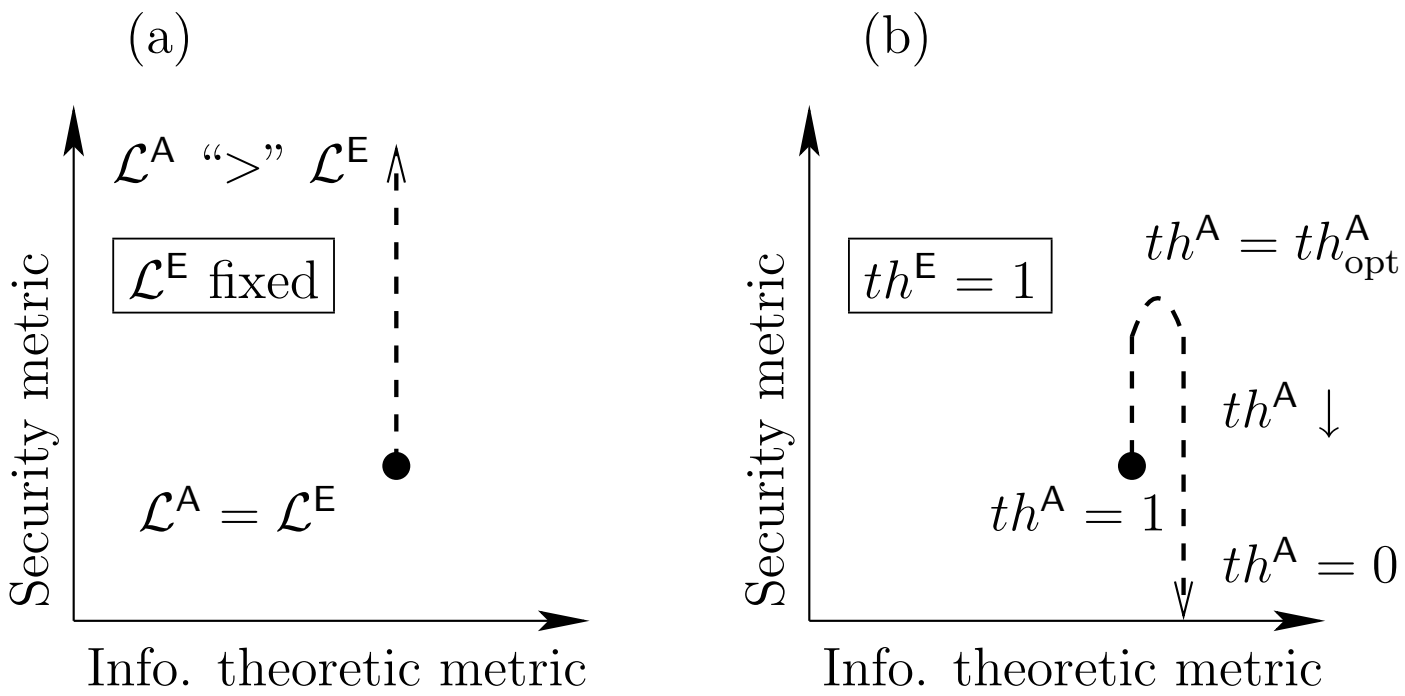


Figure 20: evaluation vs attack

Conclusion and perspectives

conclusion

- Put in practice "unified framework for the analysis of side-channel key recovery attacks" , on real-world measurements (DPA contest)
- Clarify the difference between evaluator and adversary: working conditions
- Side channel attacks: architecture (80%) and algorithm (20%)

perspectives

- How to combine models?
- Who will most benefit: evaluator or adversary?

Thanks for listening