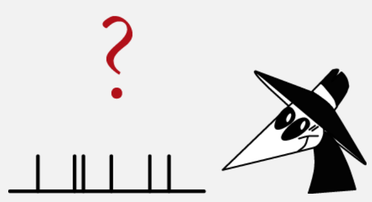
Predictive Mitigation of Timing Channels for Interactive Systems

Danfeng Zhang, Aslan Askarov and Andrew C. Myers (Cornell University)

Theoretical and empirical results show predictive mitigation of timing channels is practical for interactive systems

Timing channel threats

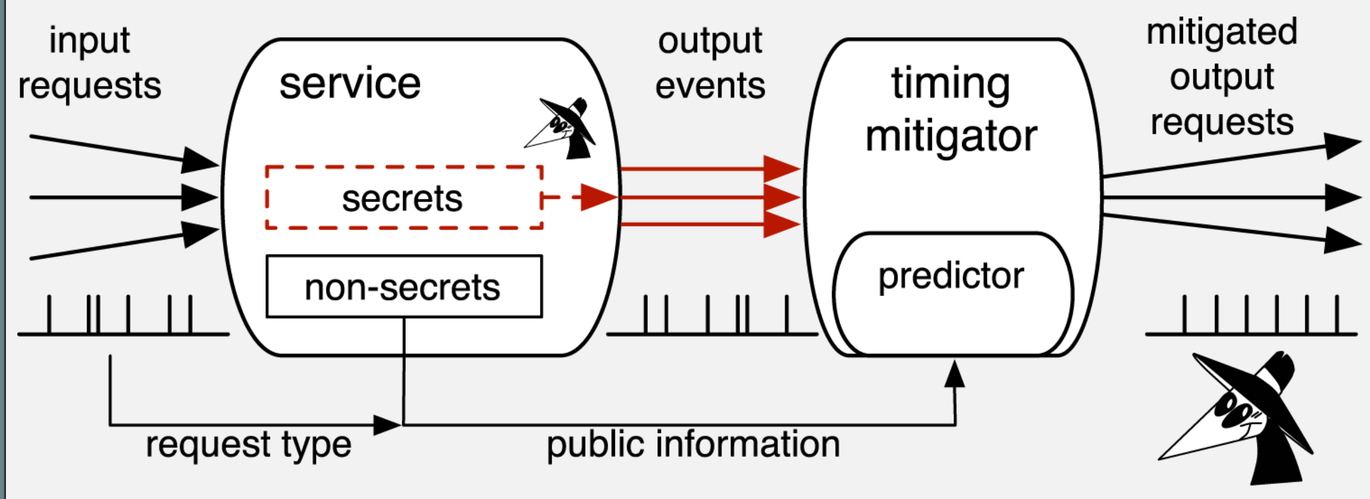
- **Side channel:** infer key from decryption time (RSA, AES)
- **Covert channel**: transmit data by controlling timing



Predictive mitigation (CCS'10)

- **Goal**
  - bound information leakage through timing channels
- **Main idea**
  - delay events according to predefined schedules
  - when events are not ready at predicted times, change to a new schedule

Interactive system model

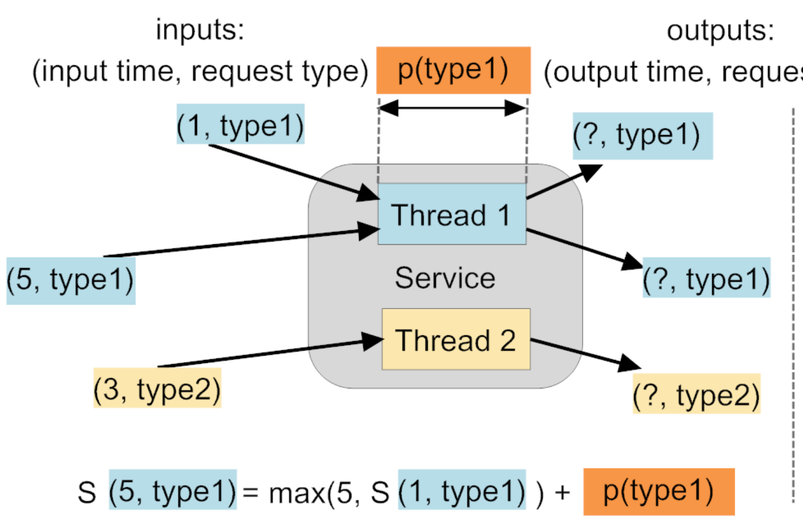


- **Attacker model**
  - attacker may influence output time
  - attacker can observe mitigated output time
- **Request type:** public payloads, e.g., URLs
- **Public information:** input time, request types

Prediction function with public information

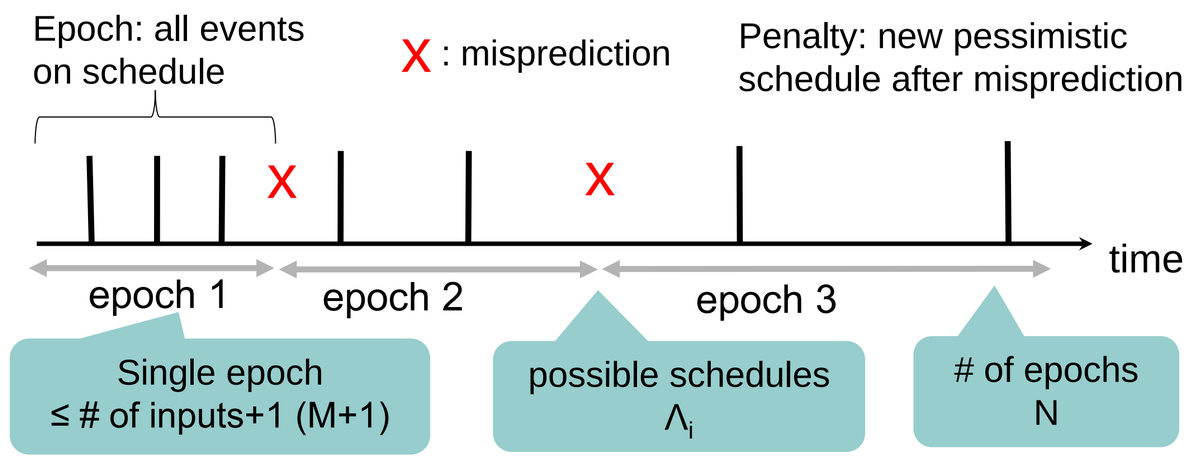
Time of outputs is predicted by public information

Thread/request type model



Leakage analysis

Idea: bound possible # of observations

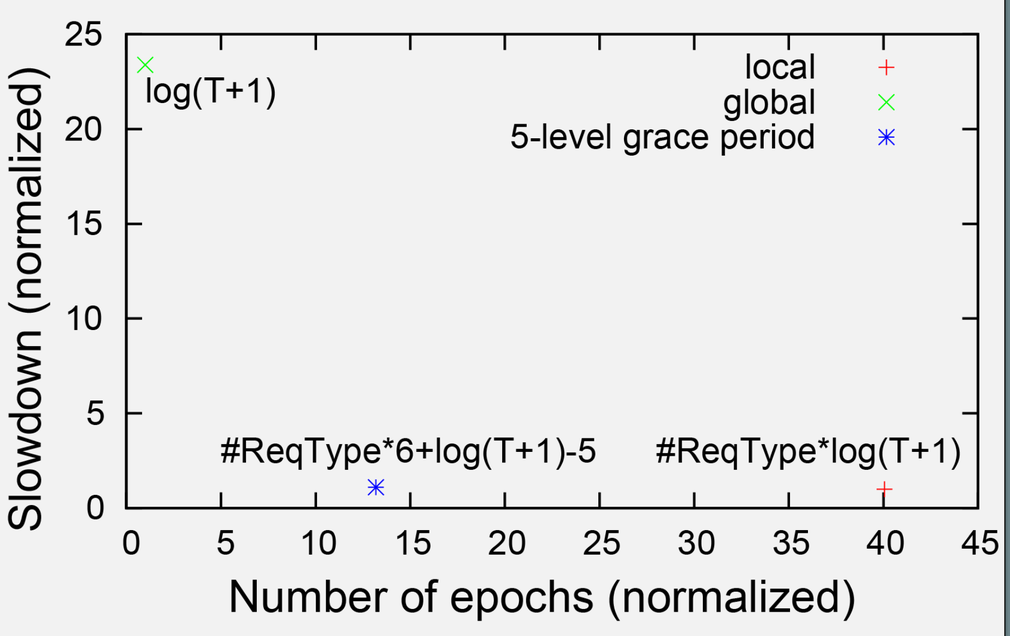


Variation ≤ $(M+1)^N \times \Pi\Lambda_i$

⟹ Leakage in bits: $N \times \log_2(M+1) + \Sigma\log_2(\Lambda_i)$

Penalty policies (bound on N)

When request type 1 has a misprediction, do we penalize request type 2?



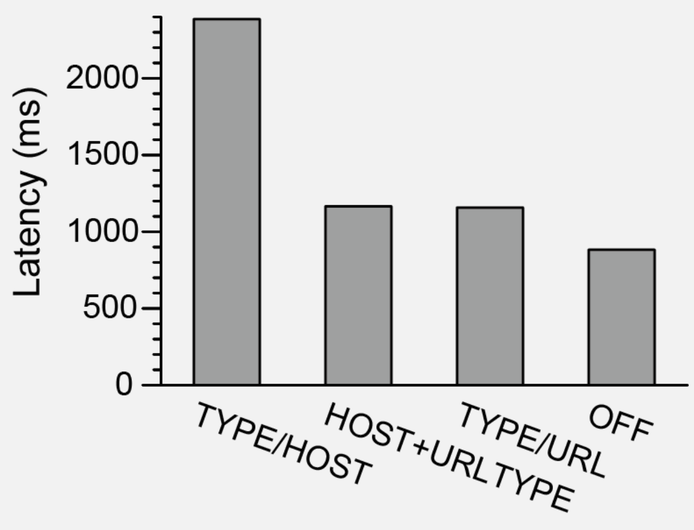
Local: only type 1 is penalized
Global: both type 1 and 2 are penalized
5-level grace period: penalize type 2 only when # of type 2's mispredictions is greater than 5

Intuition: request types with few mispredictions should receive little penalty since they leak little information

Results

Performance

- HTTP(S) proxy server that mitigates MIT CSAIL homepage (49 URLs)
- 5-level grace period
- Various request types
  - Type/Host (2)
  - Type/URL (49)
  - Host+URL type (7)

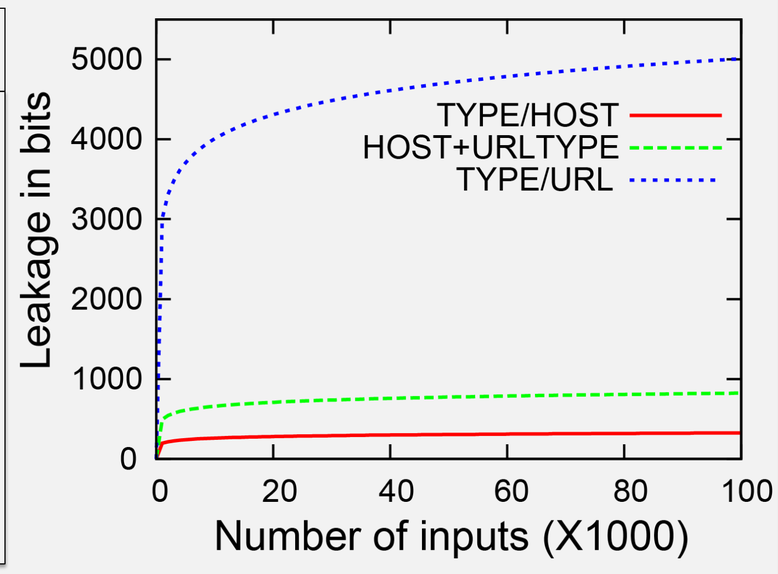


Security

Fast doubling

- start with q
- double q after misprediction
- $\Lambda_i = 1$
- Leakage bound $(6R+\log_2(T_w+1)-5)\times\log_2(M+1)$

R: # of request types
T_w: worst-case execution time (300s, the default timeout setting of Firefox)



32nd IEEE Symposium on Security and Privacy
May 22-25, 2011
Oakland, CA

Cornell University