THE OHIO STATE UNIVERSITY

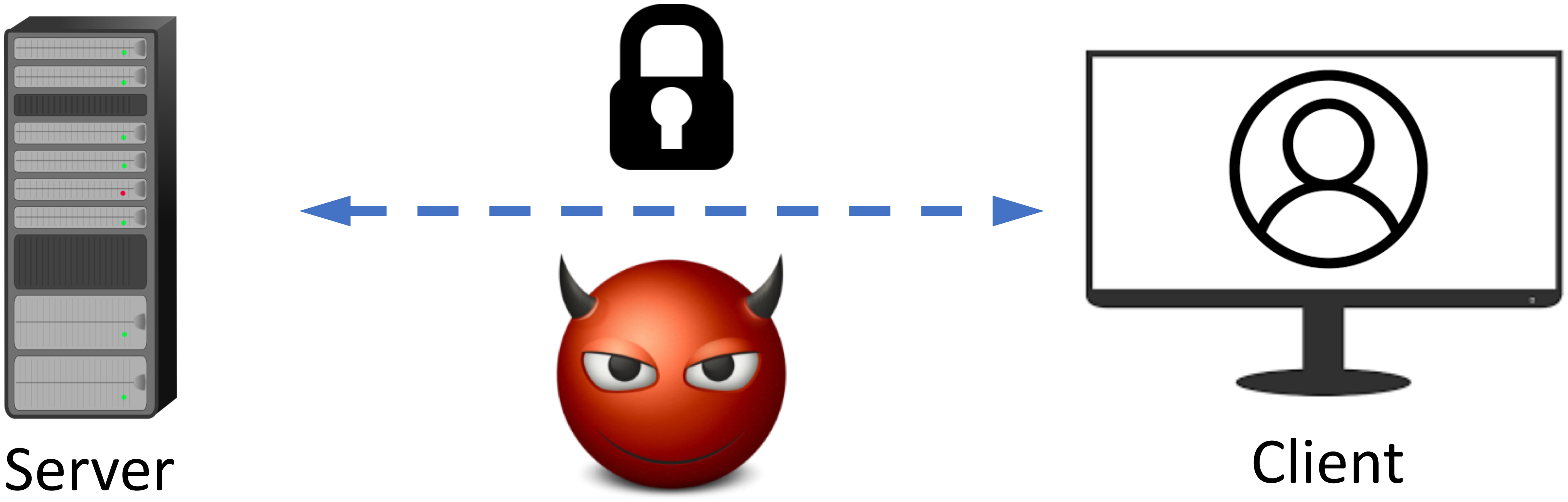
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Statistical Privacy for Streaming Traffic

Xiaokuan Zhang[1], Jihun Hamm[1], Michael K. Reiter[2], Yinqian Zhang[1]

[1]*The Ohio State University*
[2]*University of North Carolina at Chapel Hill*

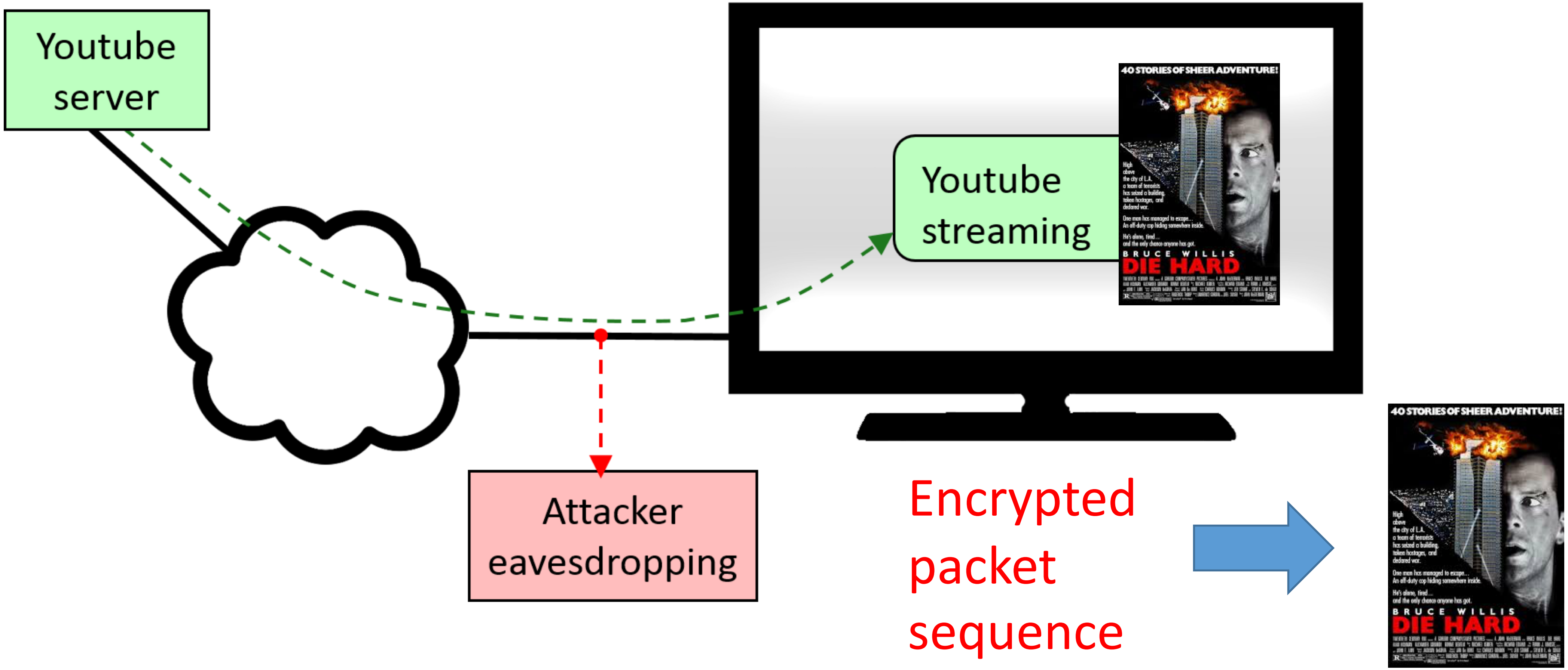
Traffic Analysis



Encrypted packet sequence → Feature Extraction → Size of packets, Timing of packets, … → Classification → Sensitive info

Traffic Analysis --- Video Streaming

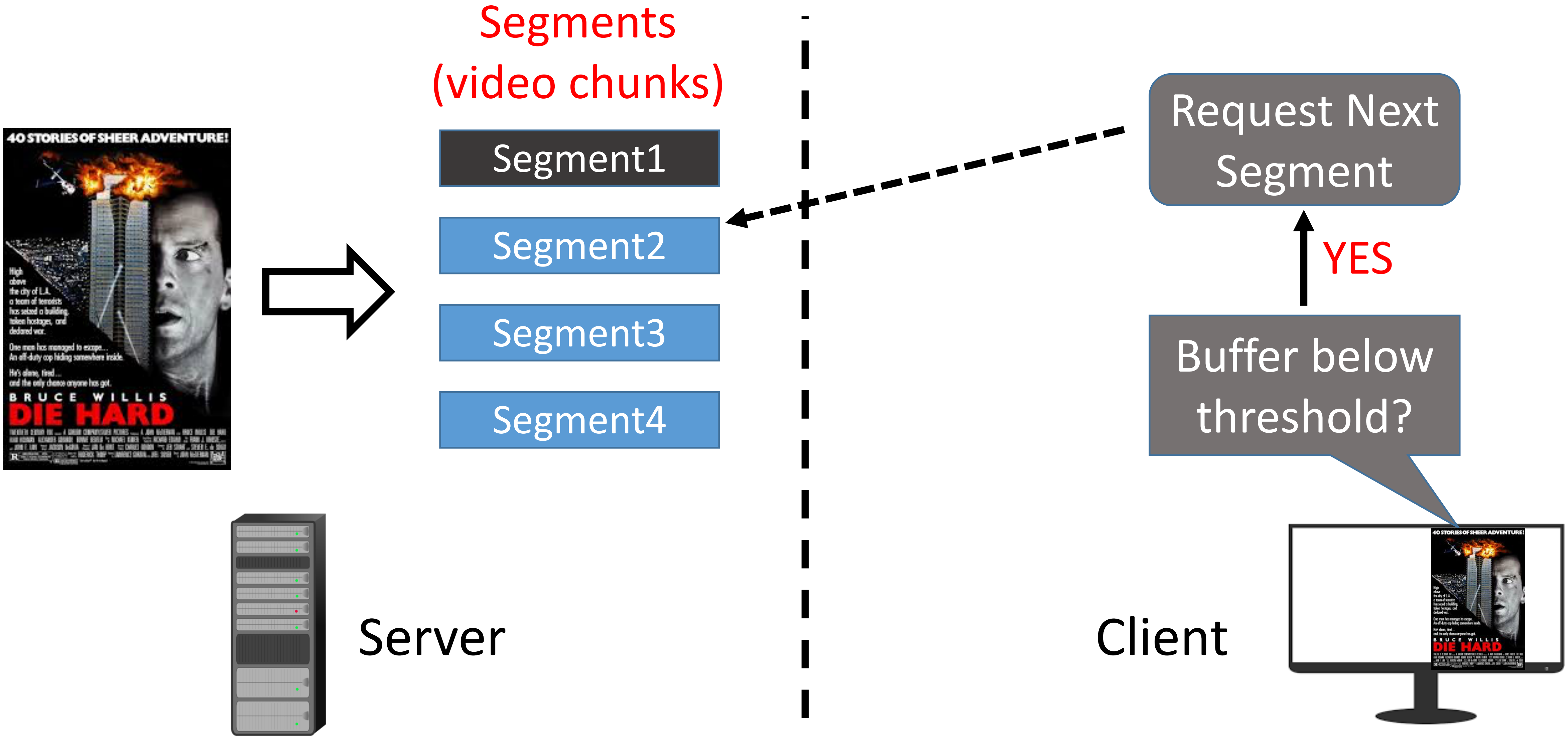
- Attacks on Encrypted Video Streams based on BURST patterns (Schuster et al. Security’17)



Schuster et al. "Beauty and the burst: Remote identification of encrypted video streams." *USENIX Security*. 2017.

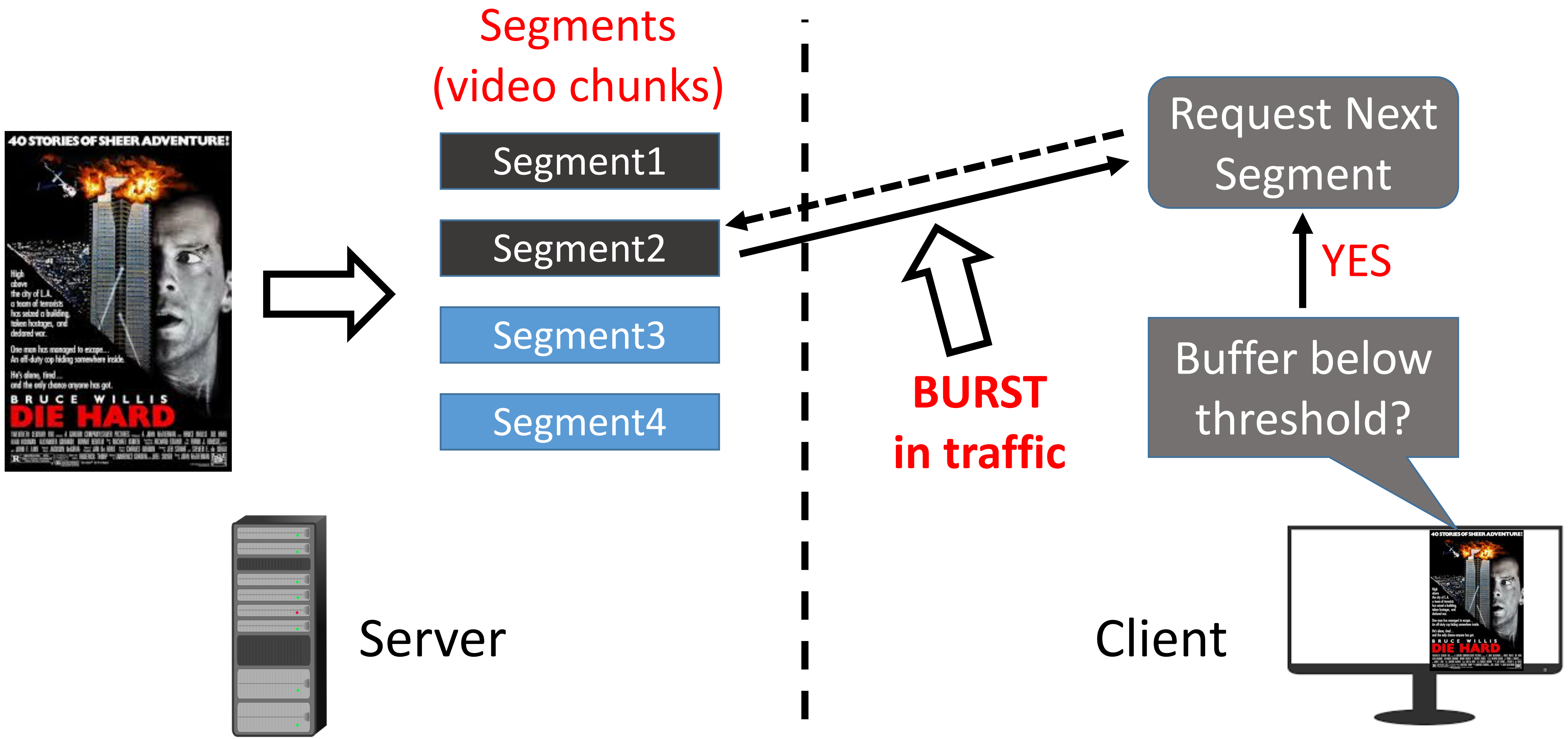
Traffic Analysis --- BURST Patterns

- MPEG-DASH standard: adaptive bitrate streaming technique



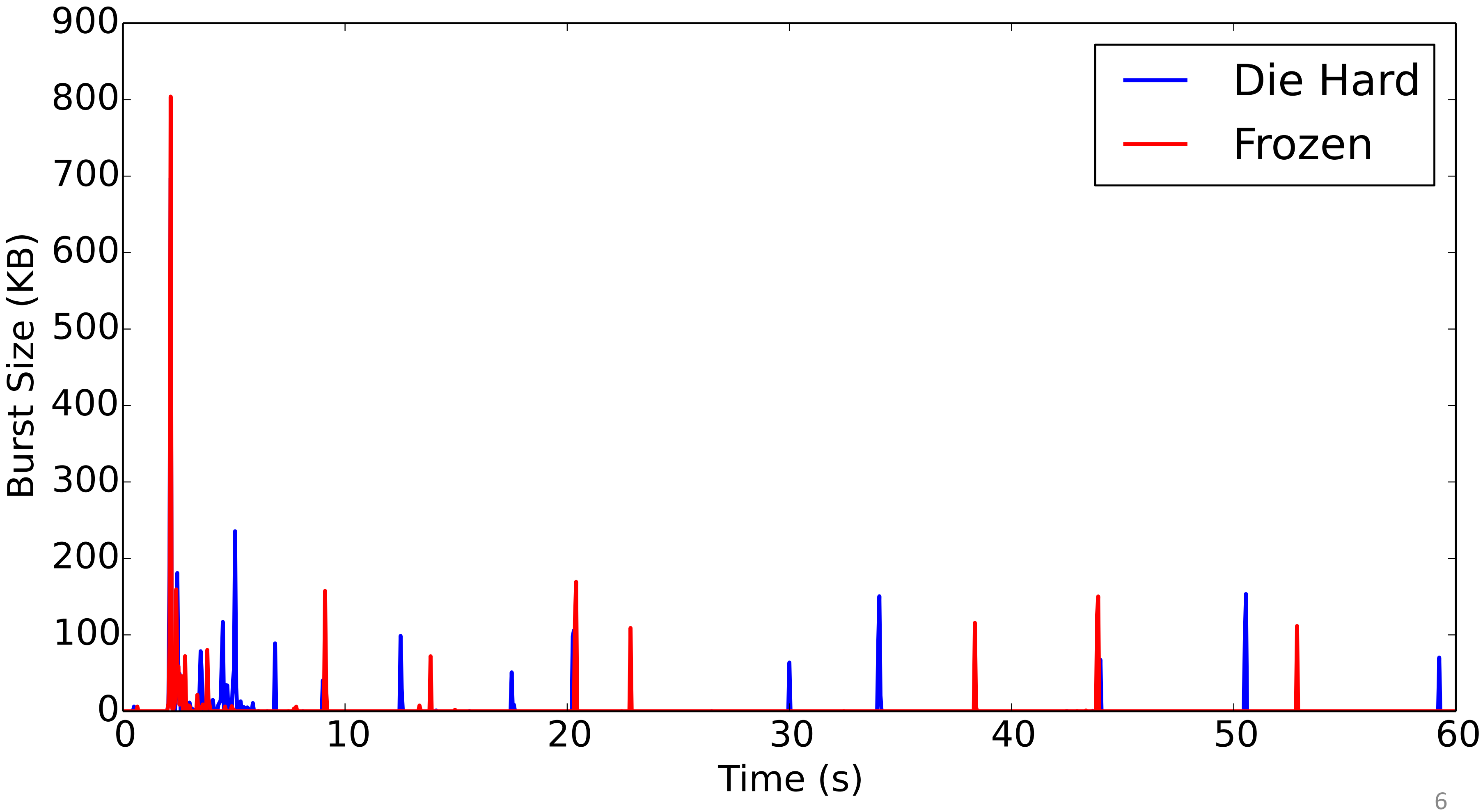
Traffic Analysis --- BURST Patterns

- MPEG-DASH standard: adaptive bitrate streaming technique



Traffic Analysis --- BURST Patterns

- Intuition: different videos have different **BURST** patterns

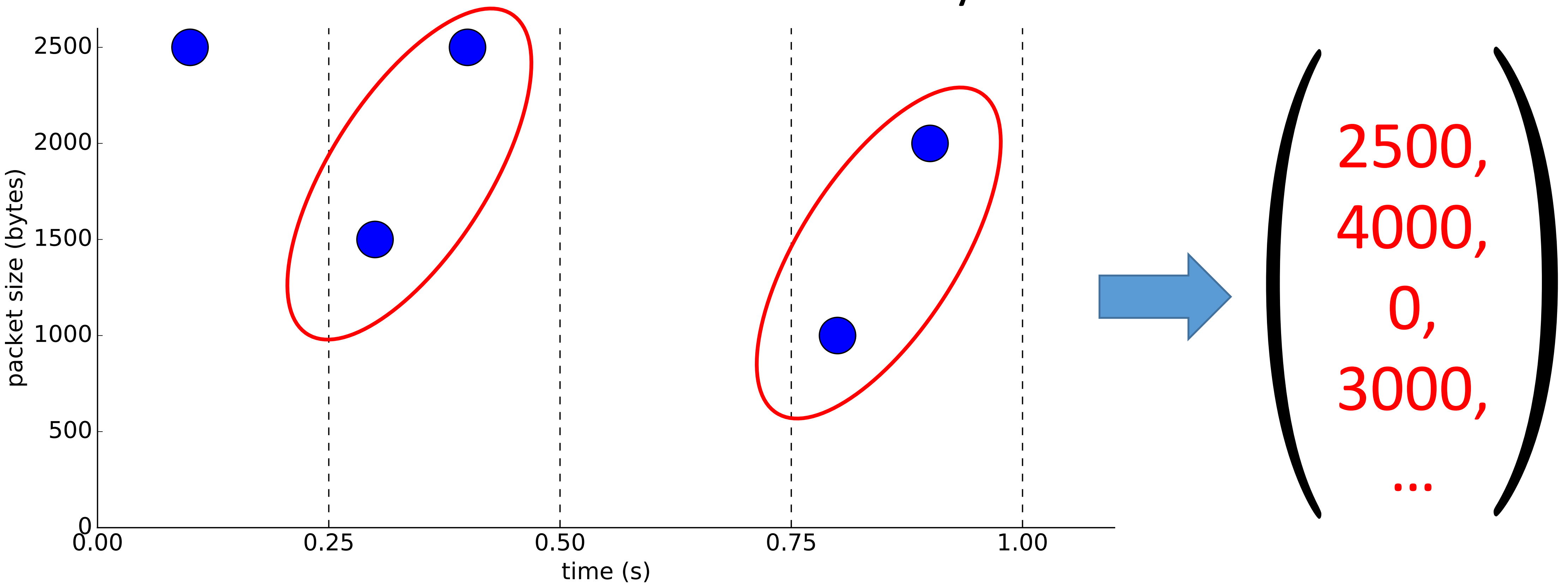


Attack Replication

- Data Collection
  - 40 videos, 100 traces per video (4000 traces)
  - Record (timestamp, packet size) of the first 3 mins
  - Automated using Selenium + Tshark

Attack Replication

- Preprocessing
  - The raw data (time series) is aggregated into 0.25-second bins
  - Each 3-minute video stream → array of 720 elements



Attack Replication

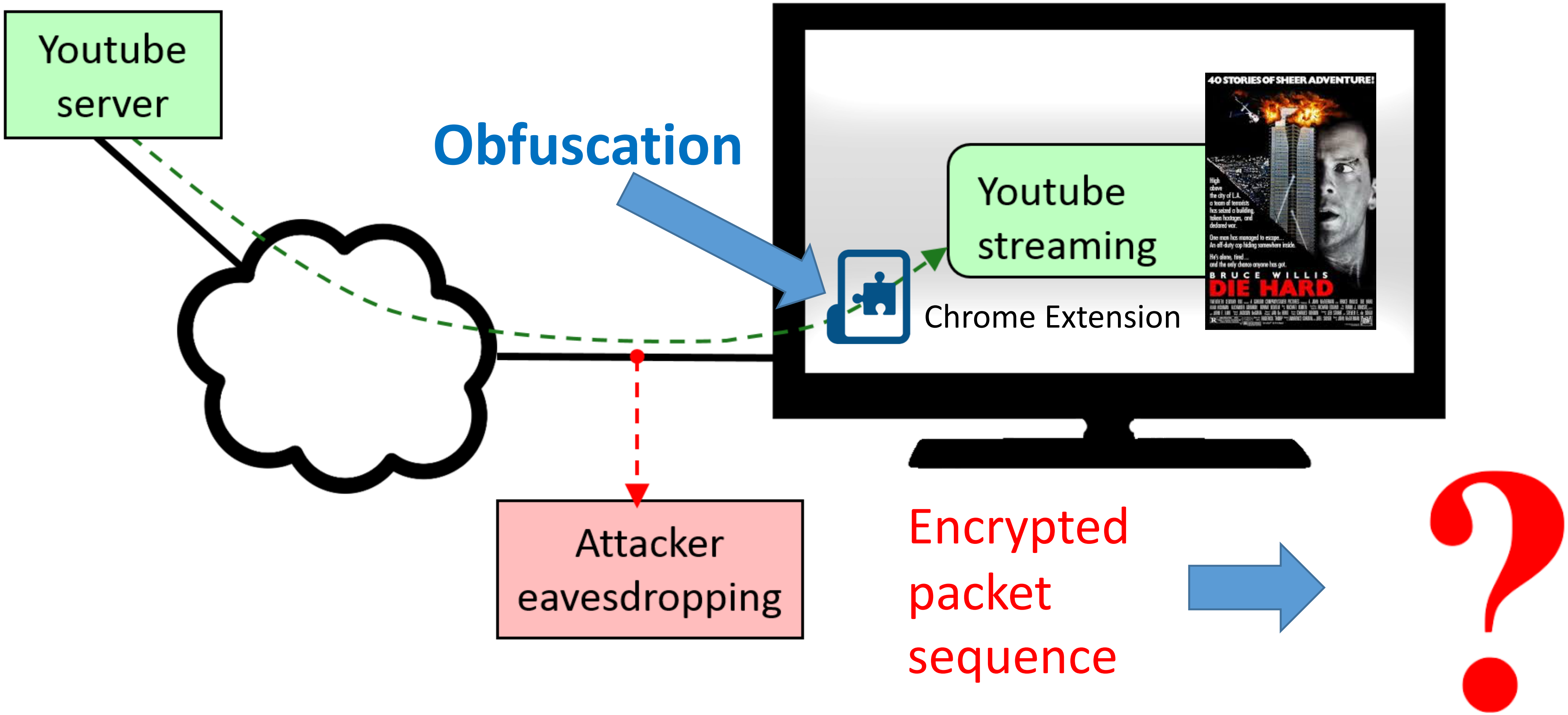
- 5 Classifiers
  - Support Vector Machine (SVM)
  - Logistic Regression (LR)
  - Random Forest (RF)
  - Neural Net
  - Convolutional Neural Net (CNN)

- Classification Result (5-fold cross-validation)

Model	SVM	LR	RF	Neural Net	CNN
Average Accuracy	0.809	0.823	0.751	0.831	0.944
Standard Deviation	0.067	0.063	0.046	0.011	0.004

Traffic Analysis --- Our Work

- Our work: defense using obfuscation



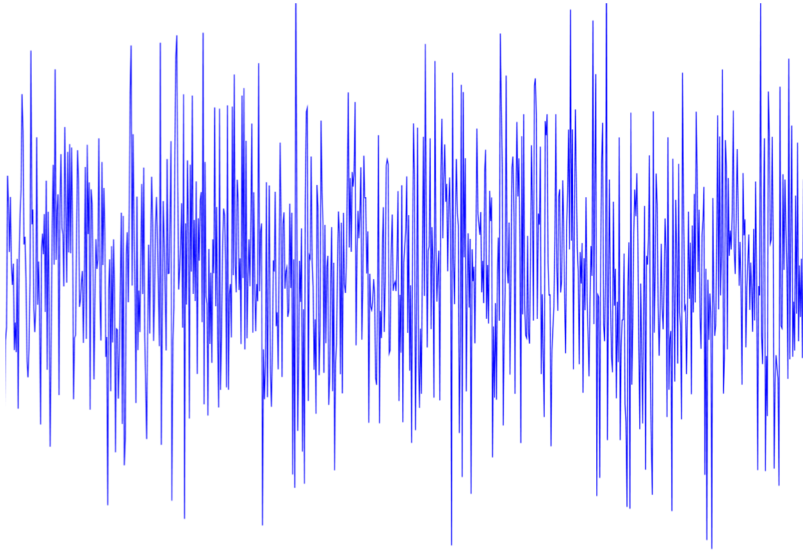
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion

Defense 1: Adversarial ML

- Defend against ML adversaries
- Crafting Adversarial Samples
  - Fast Gradient Sign Method (FGSM)

+



=

$$\eta \, \mathrm{sign}(\nabla_x L(g(x;\theta), y))$$

Defense 1: Adversarial ML

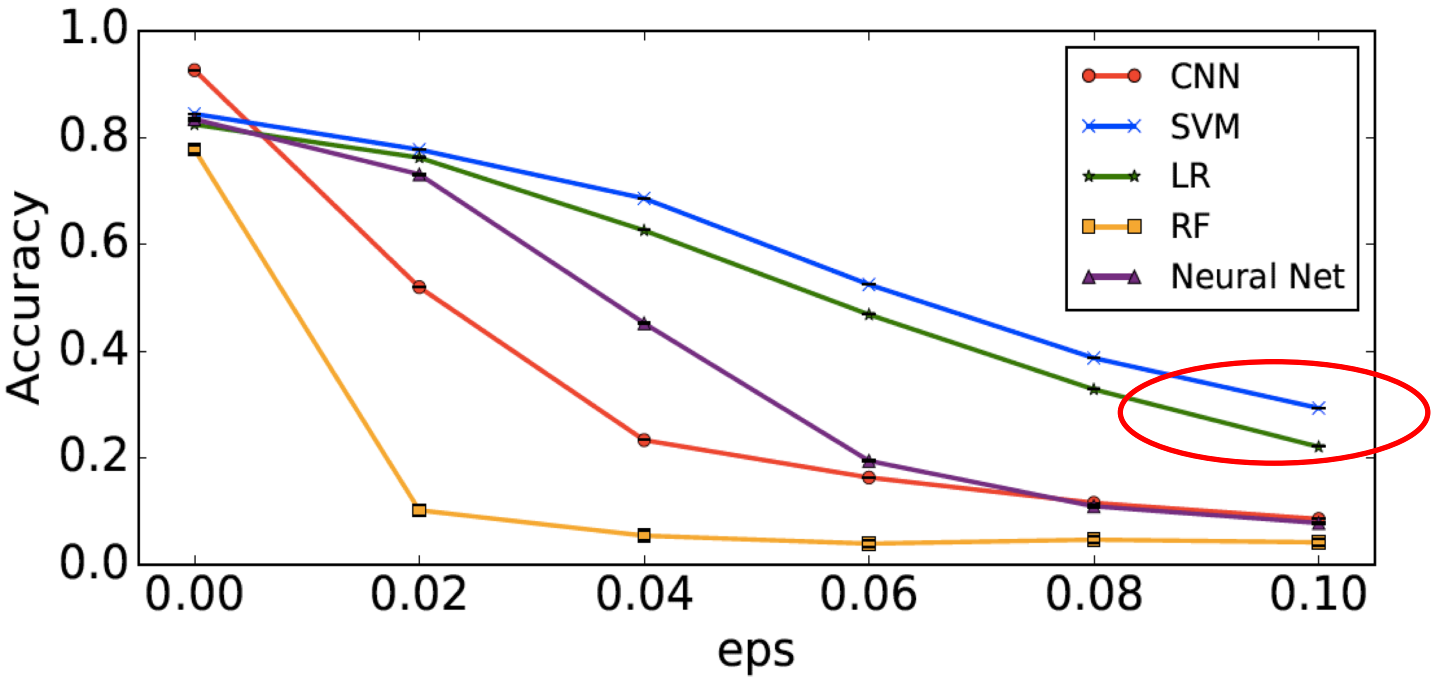
- Targets the CNN (eps=0.1): 0.944 -> 0.086
- Limitations of Adversarial Samples

Attacker may choose a different classifier

Attacker may conduct adversarial training (0.086 → 0.908)

Not so effective against others!

More **principled** approach?



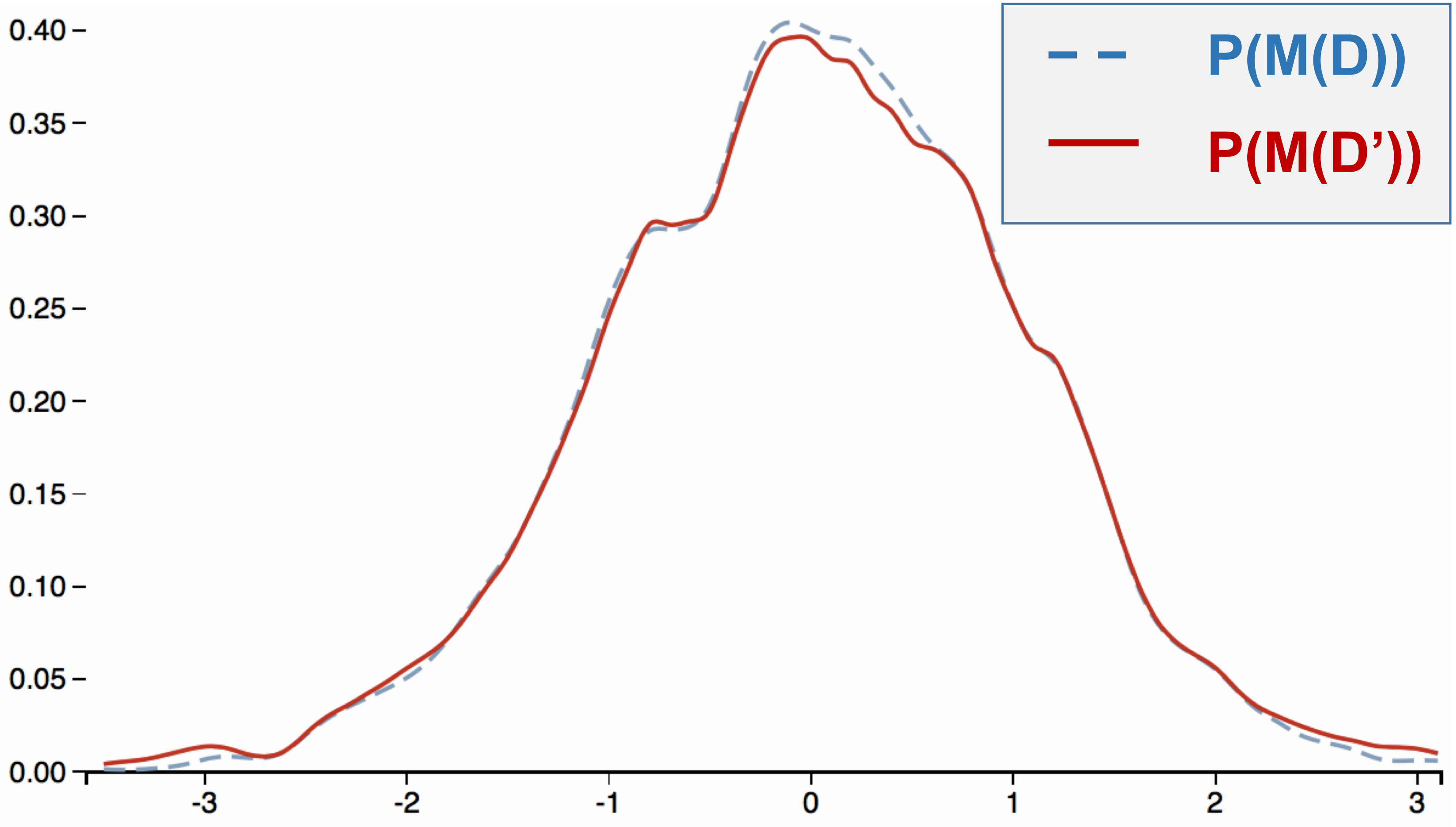
Outline

Defense 2: Differential Privacy

- Privacy in database
- Adding noise with a randomized Alg. M

D
Alice
Bob

D’
Bob



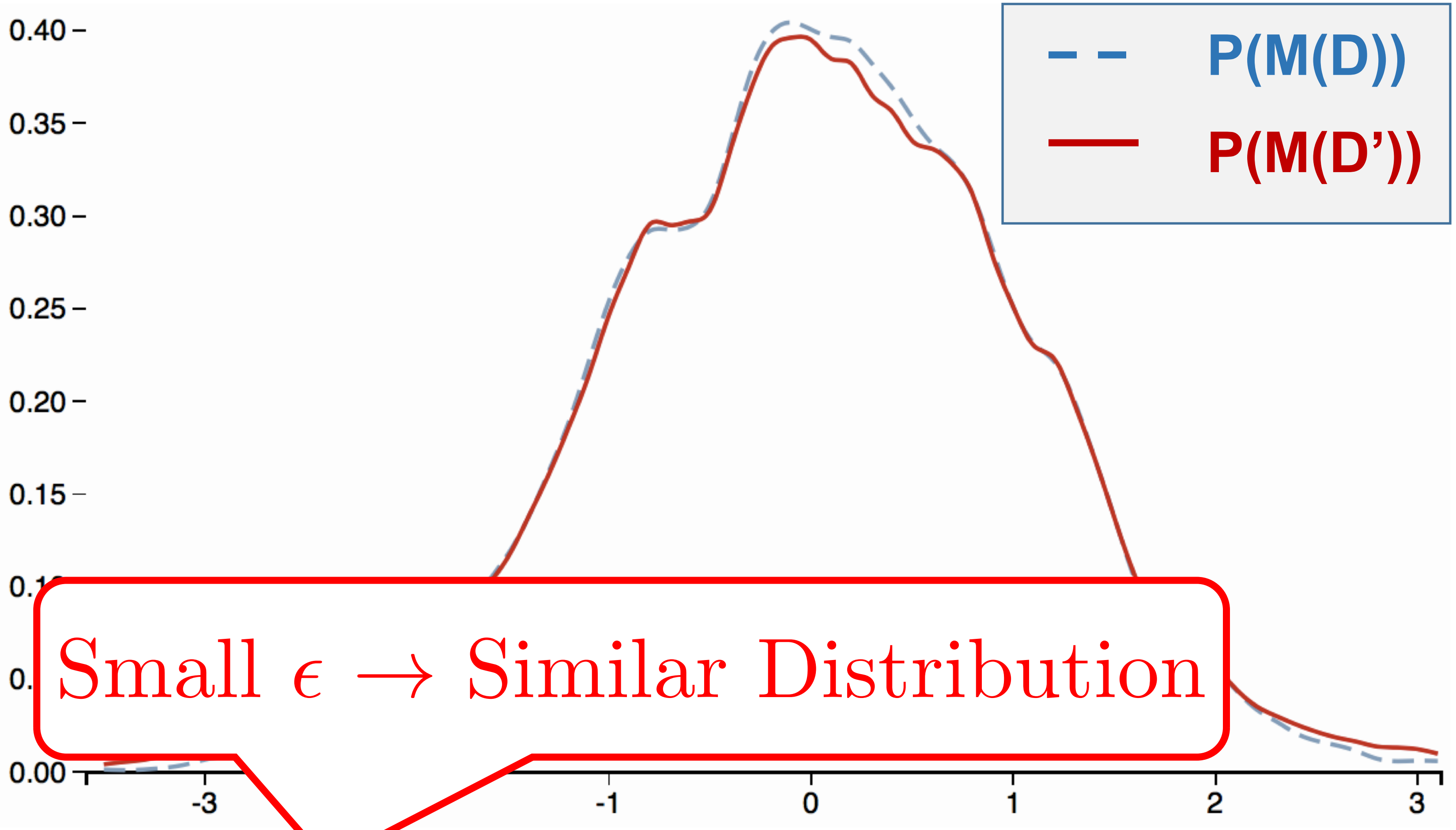
$$P(M(D) = s) \leq exp(\epsilon) \times P(M(D') = s)$$

Defense 2: Differential Privacy

- Privacy in database
- Adding noise with a randomized Alg. M

D
Alice
Bob

D'
Bob



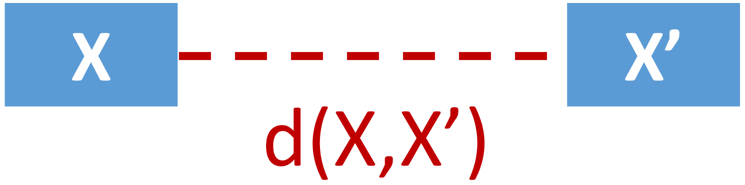
Small ϵ → Similar Distribution

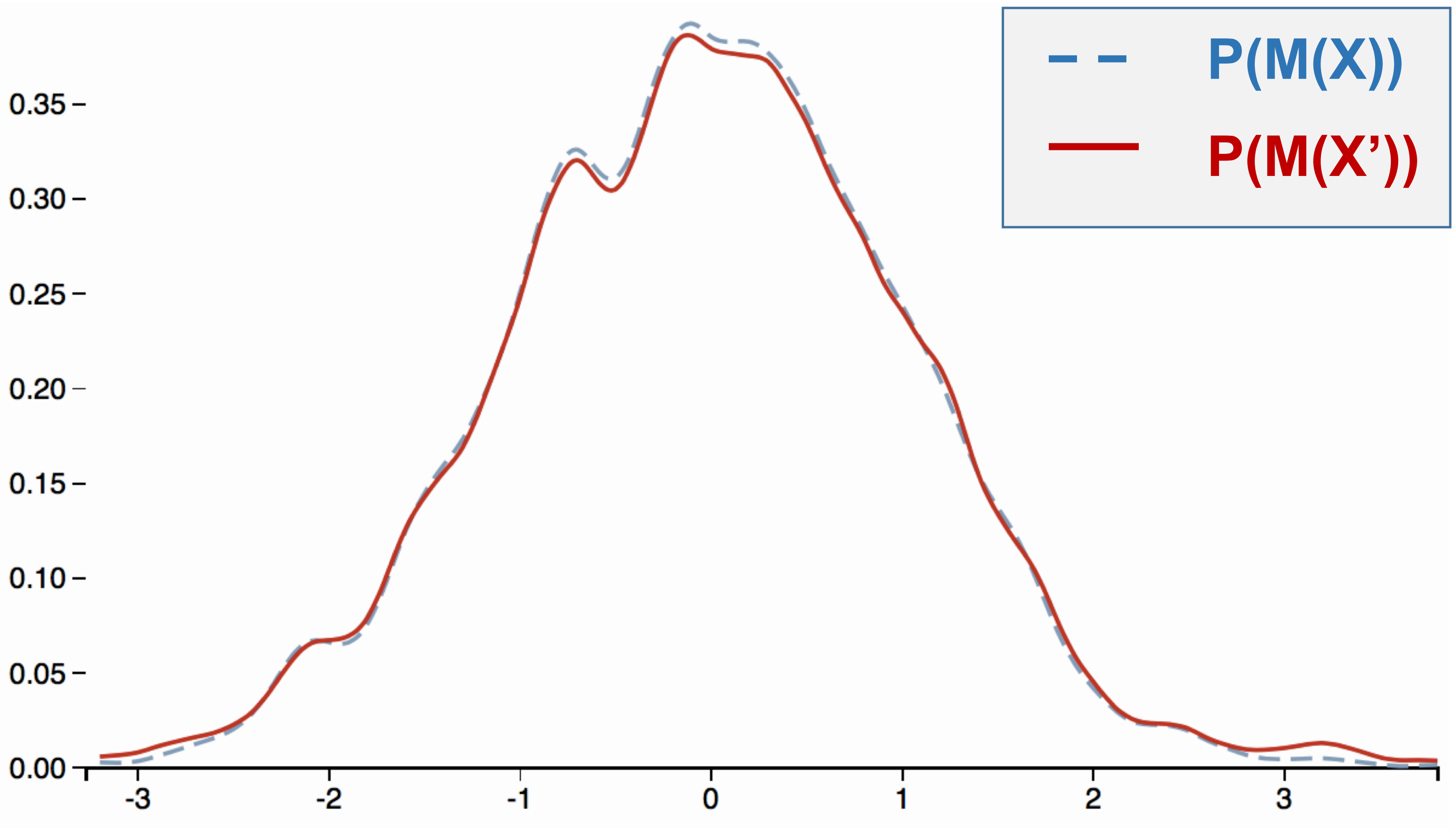
$$P(M(D) = s) \leq exp(\epsilon) \times P(M(D') = s)$$

Defense 2: Differential Privacy --- d-privacy

Calculating randomized results from data object

Parameterizing the indistinguishability with distance metric d



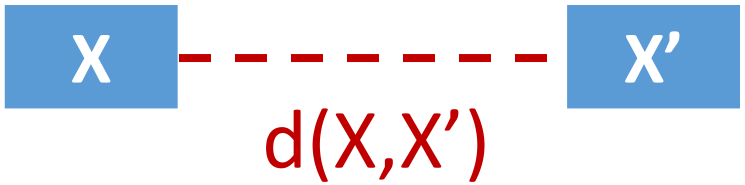


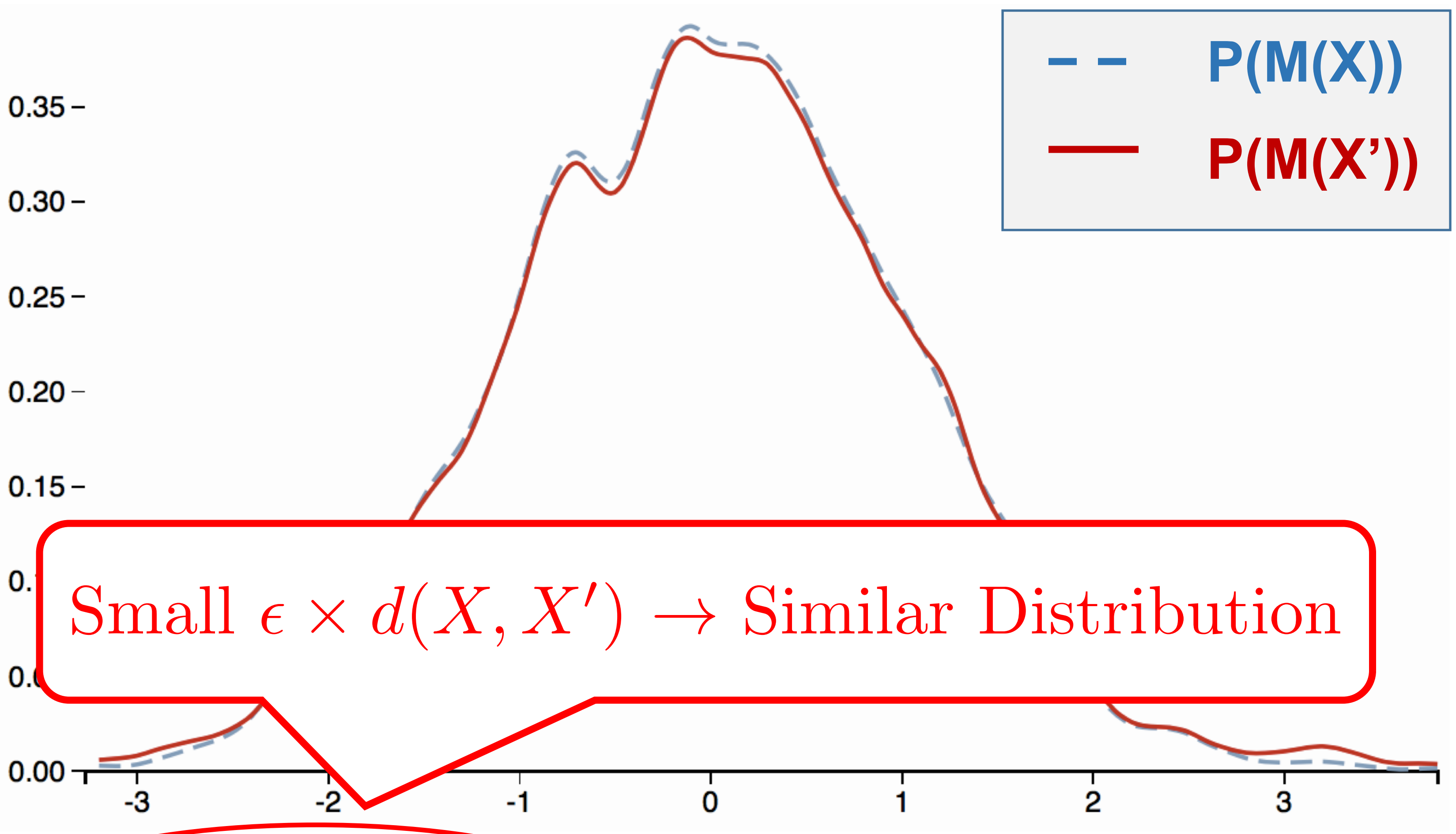
$$P(M(X) = s) \leq exp(\epsilon \times d(X, X')) \times P(M(X') = s)$$

Defense 2: Differential Privacy --- d-privacy

Calculating randomized results from data object

Parameterizing the indistinguishability with distance metric d





$$P(M(X) = s) \leq exp(\epsilon \times d(X, X')) \times P(M(X') = s)$$

Defense 2: Differential Privacy --- FPA_k & d*

- Fourier Perturbation Algorithm (FPA_k): Rastogi et al. (SIGMOD'10)

  $FPA_k(Q,\lambda)$ is ϵ-differentially private for $\lambda = \sqrt{k}\Delta_2(\mathbb{Q})/\epsilon$,

  $\Delta_2(\mathbb{Q})$ denotes the L2 sensitivity of a set of Qs.

- d*-private Mechanism: Xiao et al. (CCS'15)

$$d^*(x, x') = \sum_{i \geq 1} |(x[i] - x[i-1]) - (x'[i] - x'[i-1])|$$

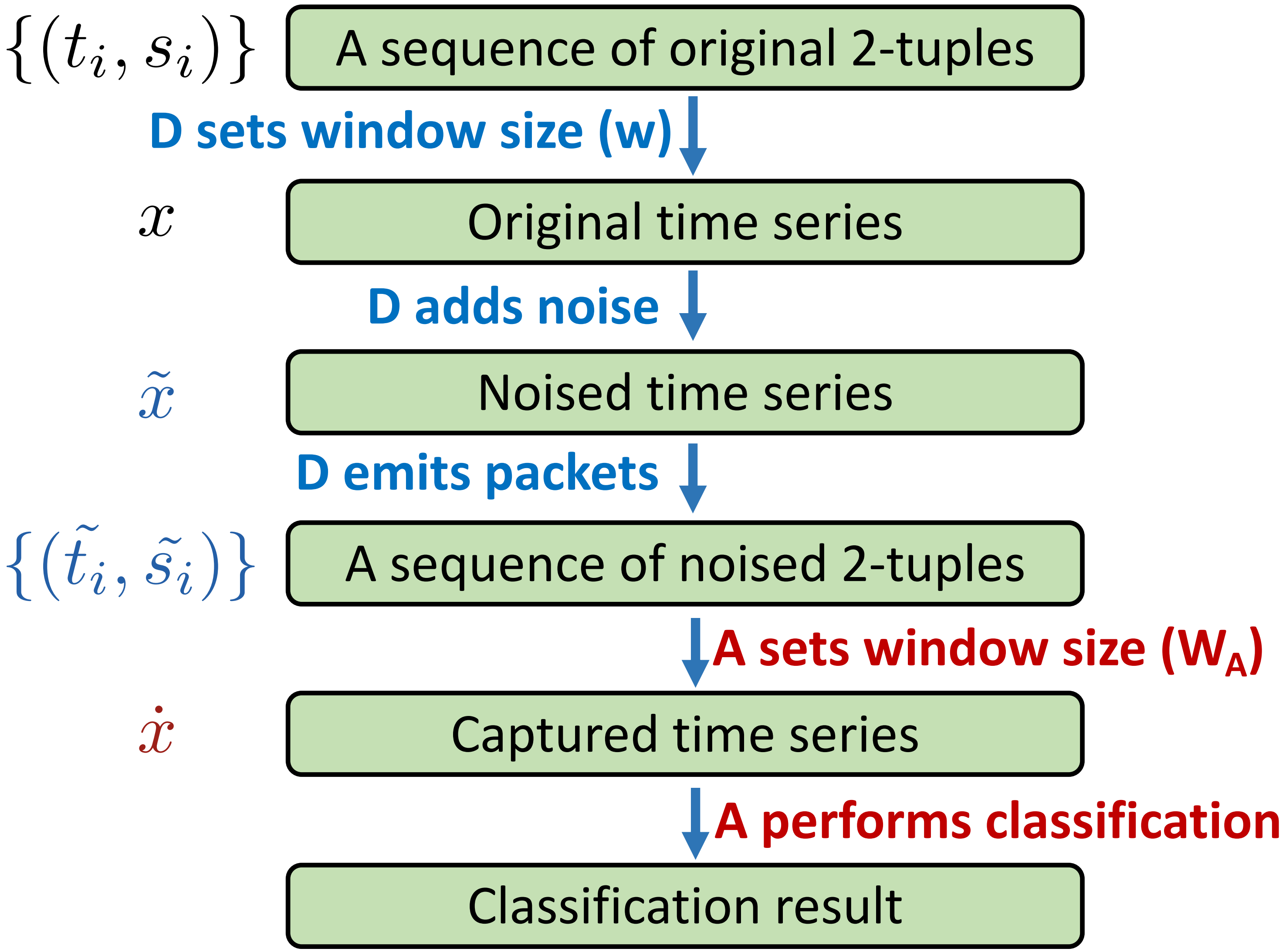
d^*-private mechanism is $(d^*, 2\epsilon)$-private and $(l_1, 4\epsilon)$-private.

Rastogi et al. "Differentially private aggregation of distributed time-series with transformation and encryption." *SIGMOD, 2010*.
Xiao et al. "Mitigating storage side channels using statistical privacy mechanisms." CCS, 2015.

Defense 2: Differential Privacy --- data flow

A: Attacker

D: Defender



Defense 2: Differential Privacy --- data flow

A: Attacker

D: Defender

$$w = w_A$$

$\{(t_i, s_i)\}$ A sequence of original 2-tuples

D sets window size (w) ↓

x Original time series

D adds noise ↓

$\tilde{x}$ Noised time series

D emits packets ↓

$\{(\tilde{t}_i, \tilde{s}_i)\}$ A sequence of noised 2-tuples

↓ **A sets window size (W_A)**

$\dot{x}$ Captured time series

↓ **A performs classification**

Classification result

Defense 2: Differential Privacy --- data flow

A: Attacker

D: Defender

$w = w_A$

$w \neq w_A$

$\{(t_i, s_i)\}$ A sequence of original 2-tuples

$\downarrow$ **D sets window size (w)**

x Original time series

$\downarrow$ **D adds noise**

$\tilde{x}$ Noised time series

$\downarrow$ **D emits packets**

$\{(\tilde{t}_i, \tilde{s}_i)\}$ A sequence of noised 2-tuples

$\downarrow$ **A sets window size (W_A)**

$\dot{x}$ Captured time series

$\downarrow$ **A performs classification**

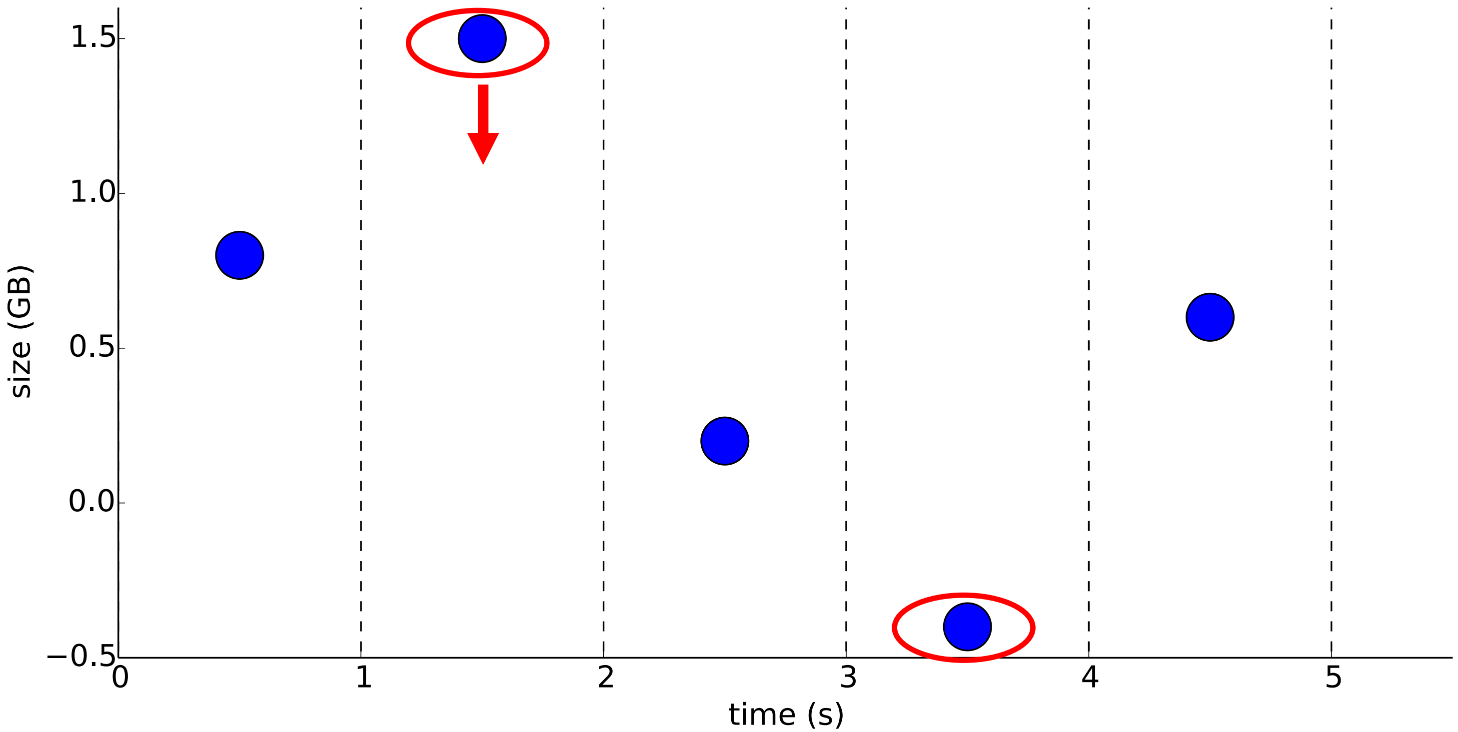
Classification result

Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion

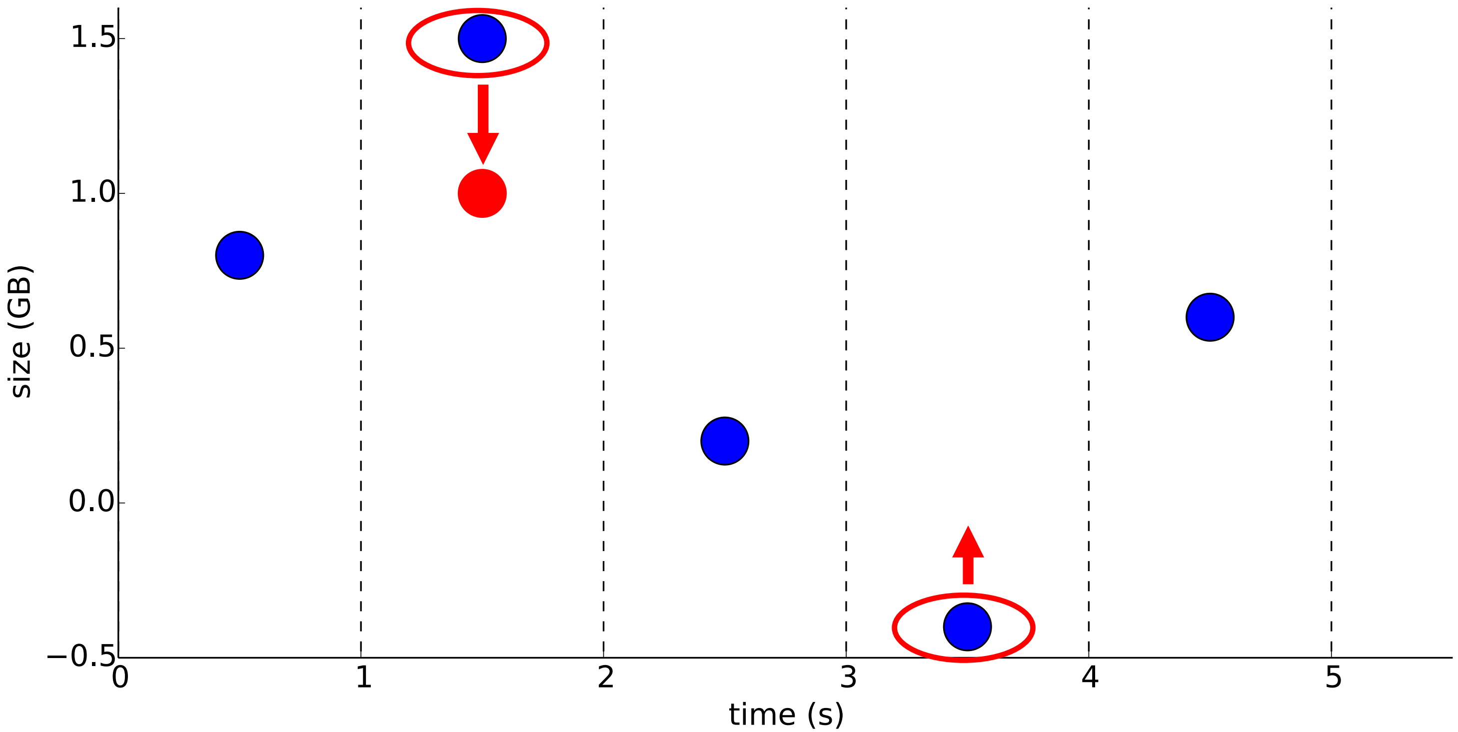
Evaluation

- 40x100 traces
- Params: $\epsilon = \{5 \times 10^{-8}, 5 \times 10^{-7}, \cdots, 50\}$
  $w = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$
- Clip bound for each window: [0, 1GB]



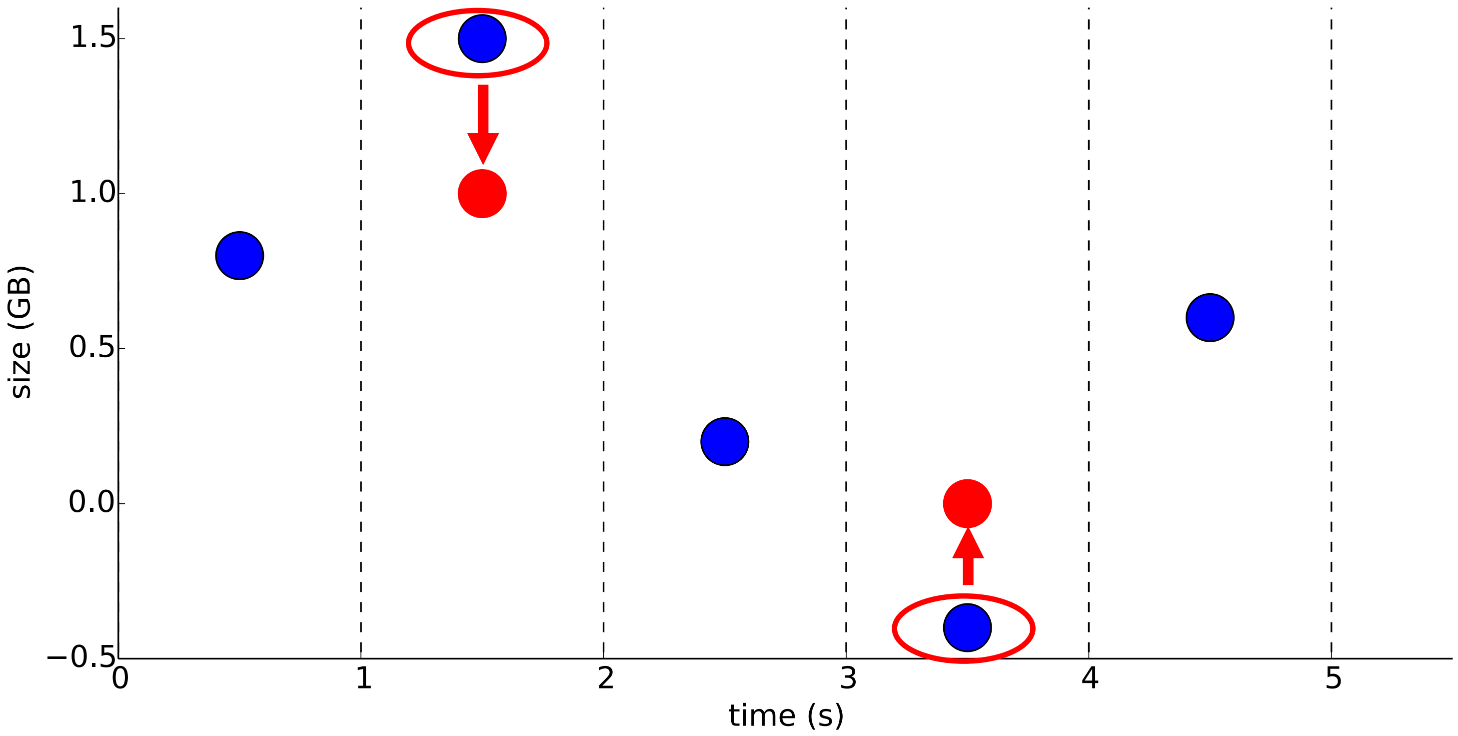
Evaluation

- 40x100 traces
- Params: $\epsilon = \{5 \times 10^{-8}, 5 \times 10^{-7}, \cdots, 50\}$
  $w = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$
- Clip bound for each window: [0, 1GB]



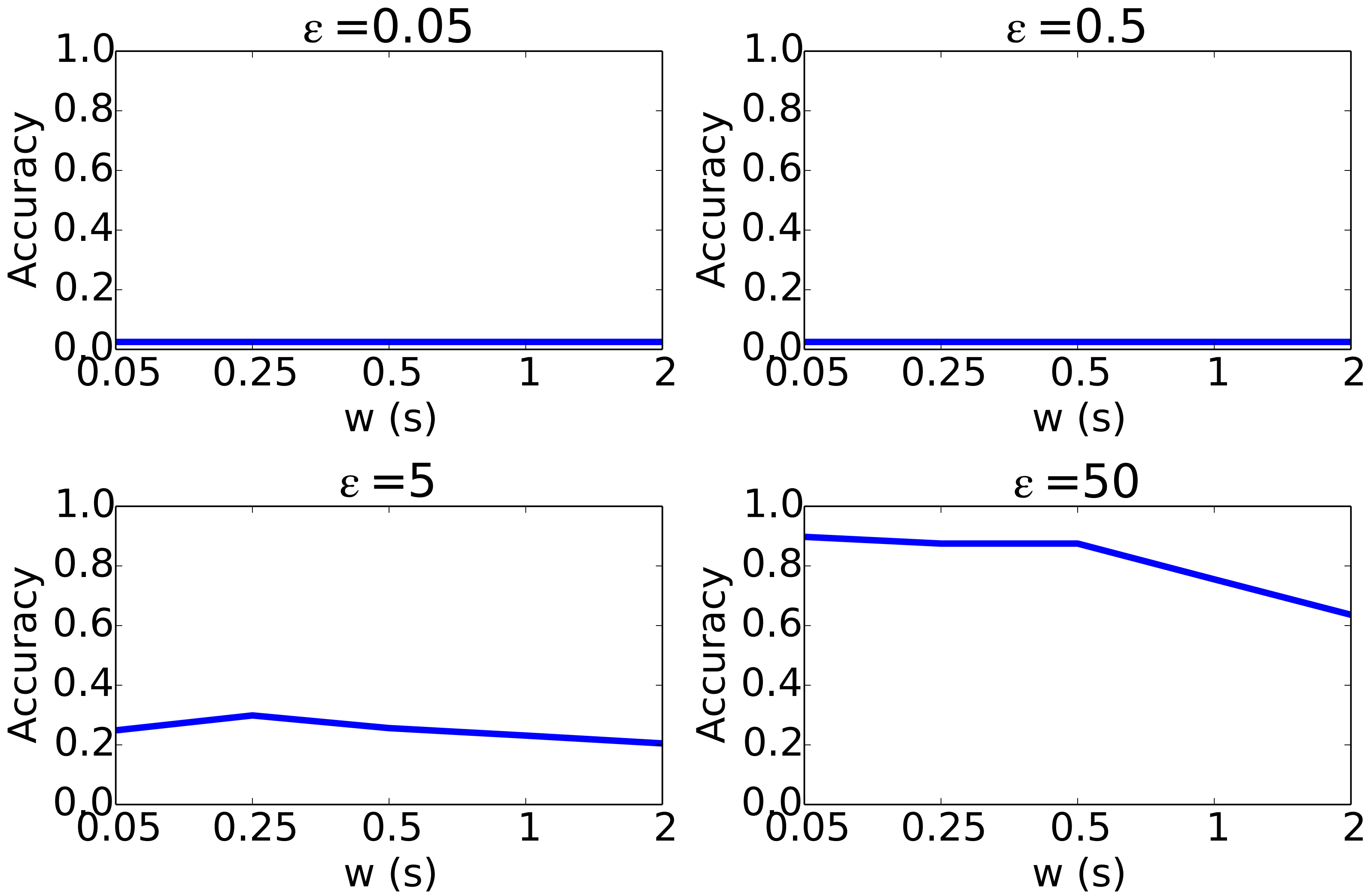
Evaluation

- 40x100 traces
- Params: $\epsilon = \{5 \times 10^{-8}, 5 \times 10^{-7}, \cdots, 50\}$
  $w = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$
- Clip bound for each window: [0, 1GB]



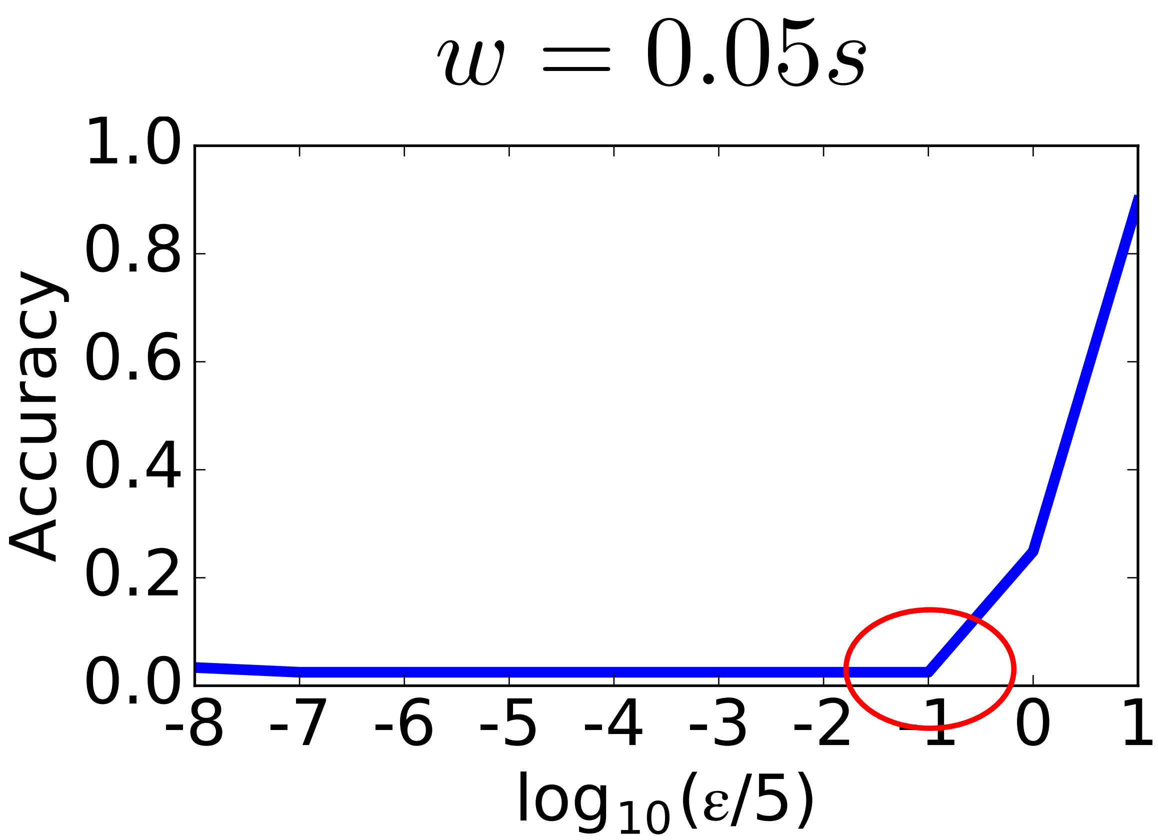
Security Evaluation --- FPA$_k$

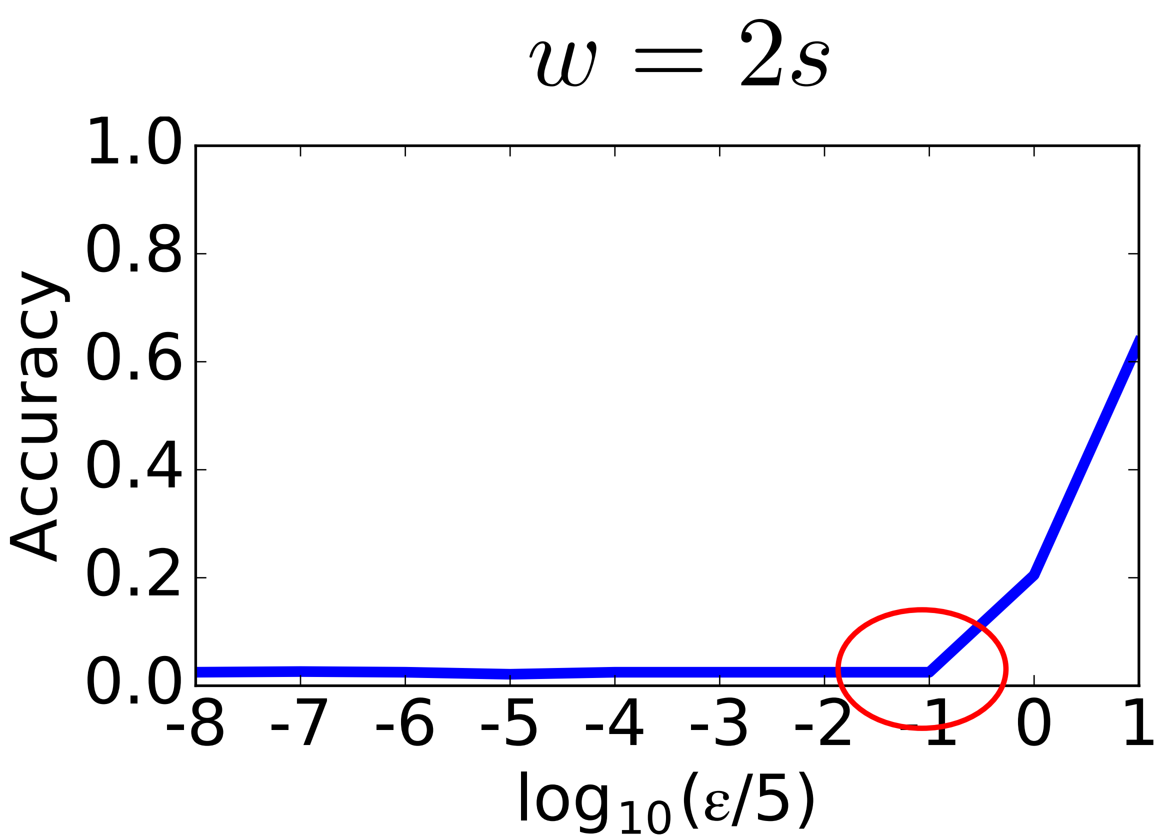
$$w_A = w\text{: effect of } w$$



Security Evaluation --- FPA_k

$$w_A = w\text{: effect of } \epsilon$$

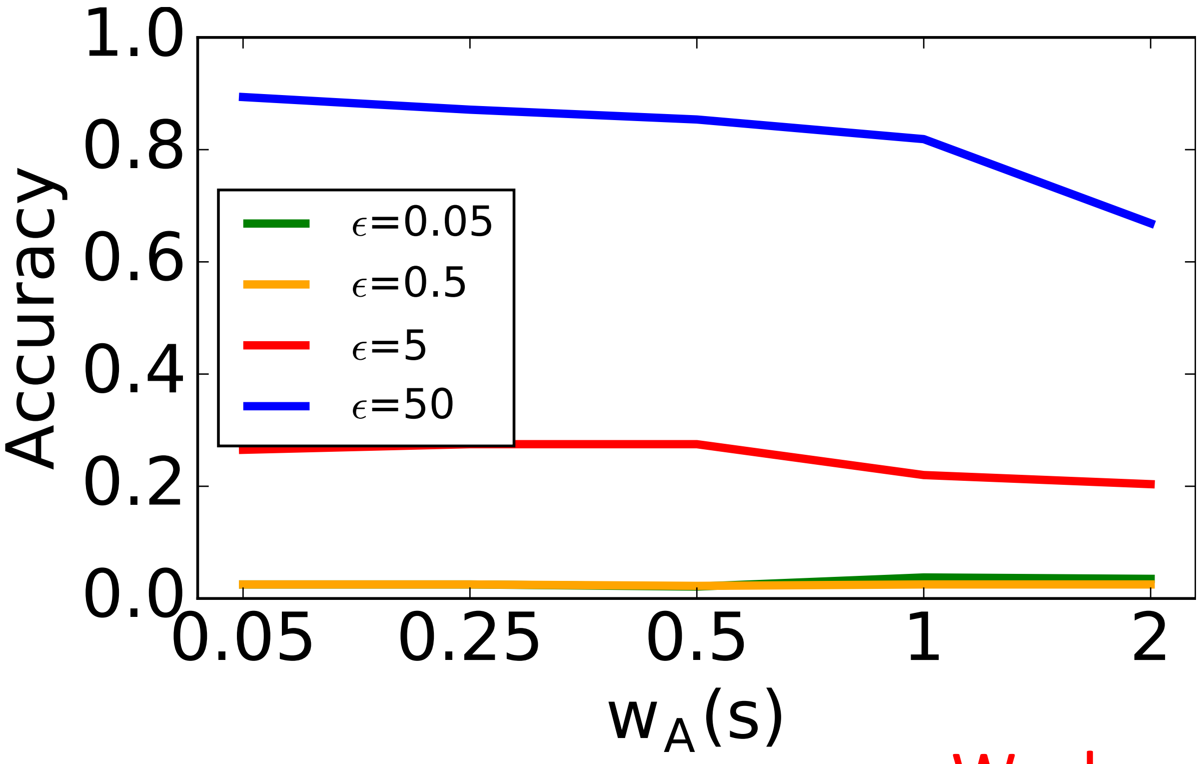




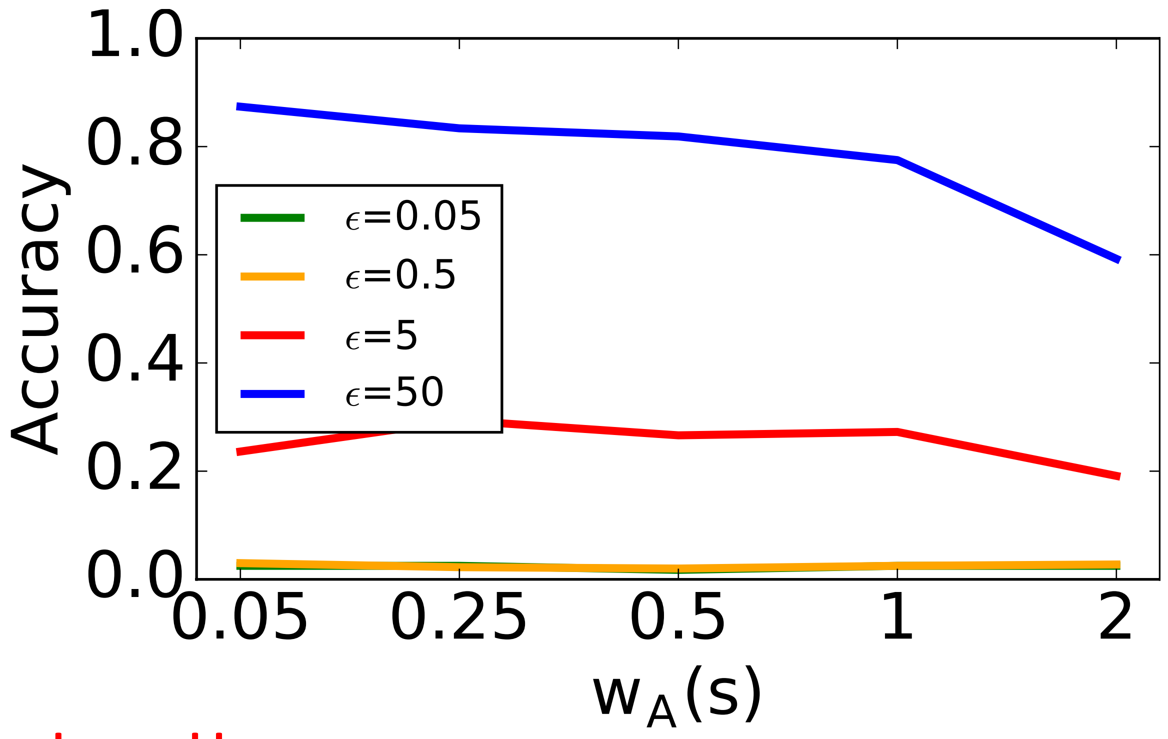
Security Evaluation --- FPA_k

$$w \neq w_A$$

$$w = 0.05s$$



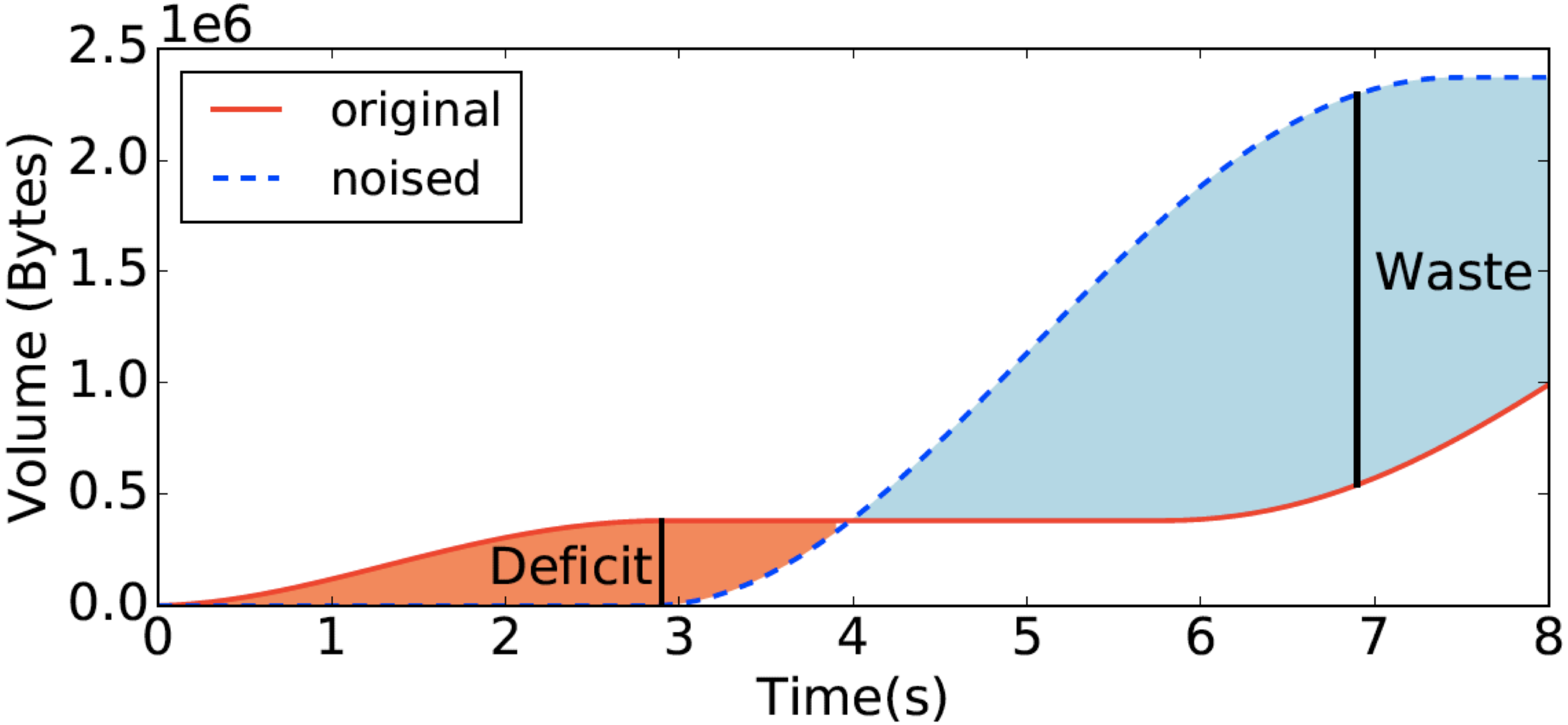
$$w = 2s$$



W_A does not matter

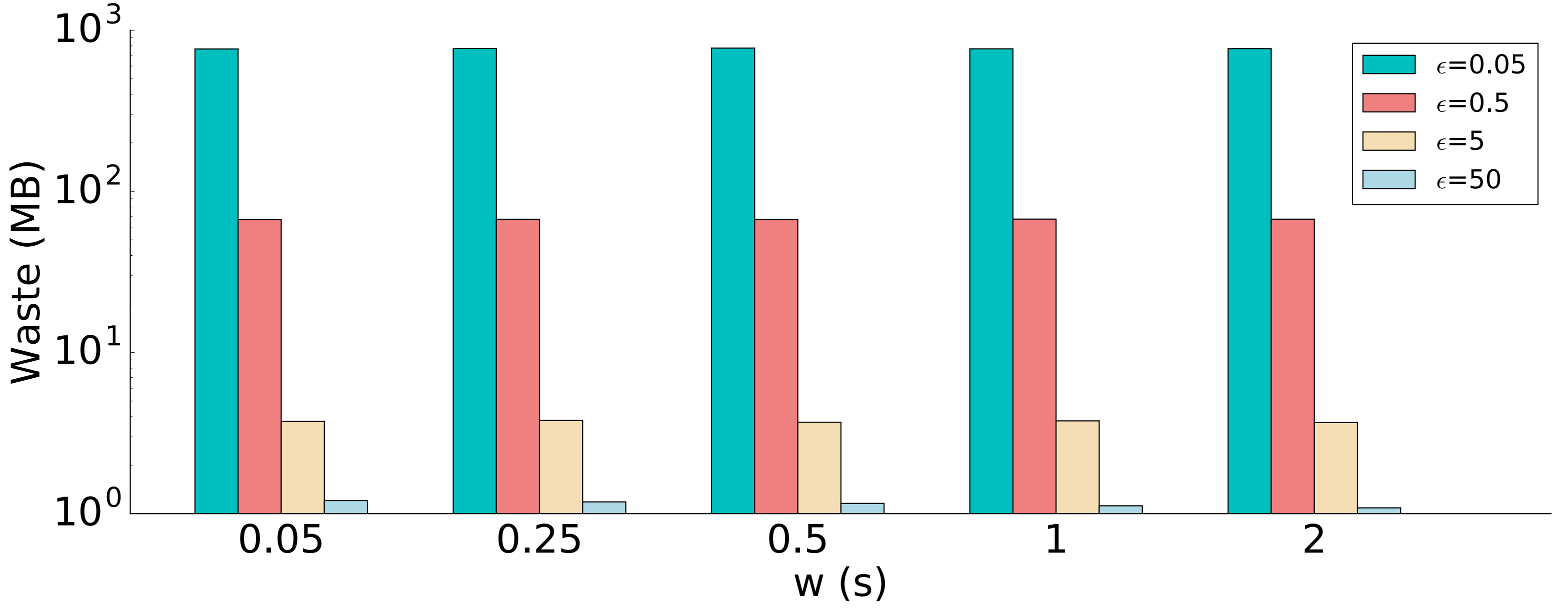
Utility Evaluation

- Original cumulative trace A, noised cumulative trace B
- Waste: $waste = \max_{1 \le i \le n}\{\max(B[i] - A[i], 0)\}$
- Deficit: $deficit = \max_{1 \le i \le n}\{\max(A[i] - B[i], 0)\}$

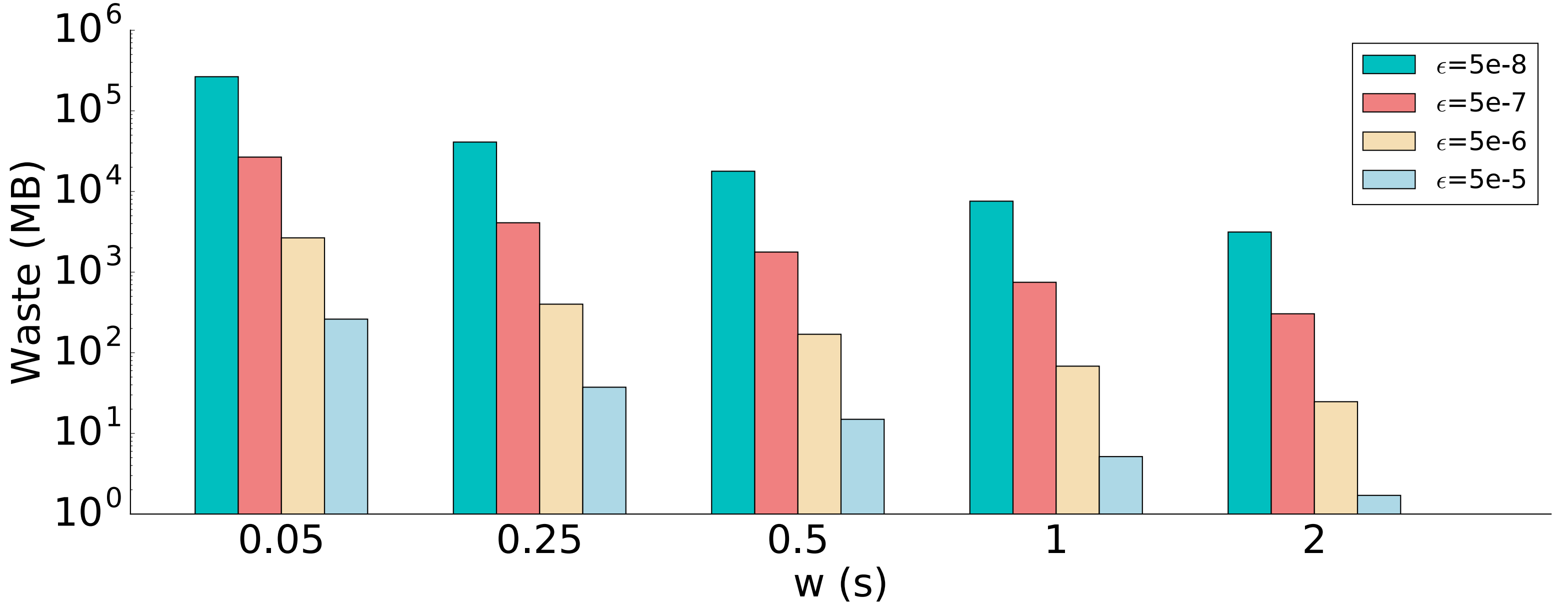


Utility Evaluation --- Waste

FPA$_k$

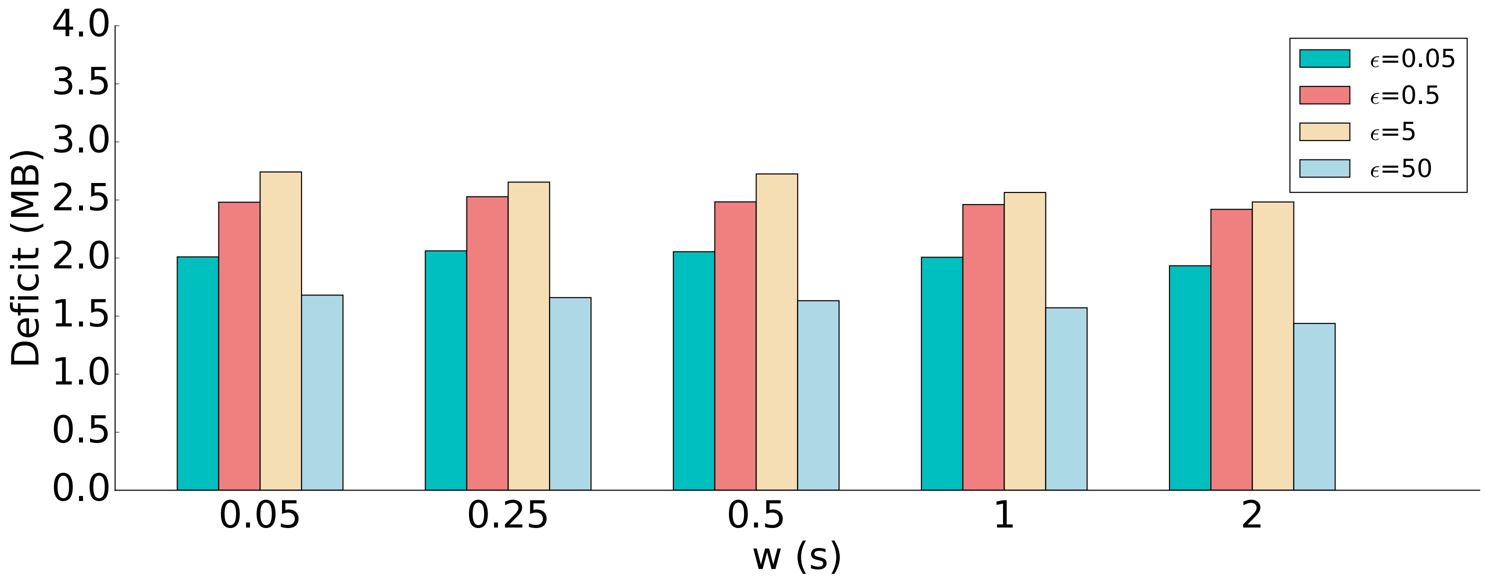


d*

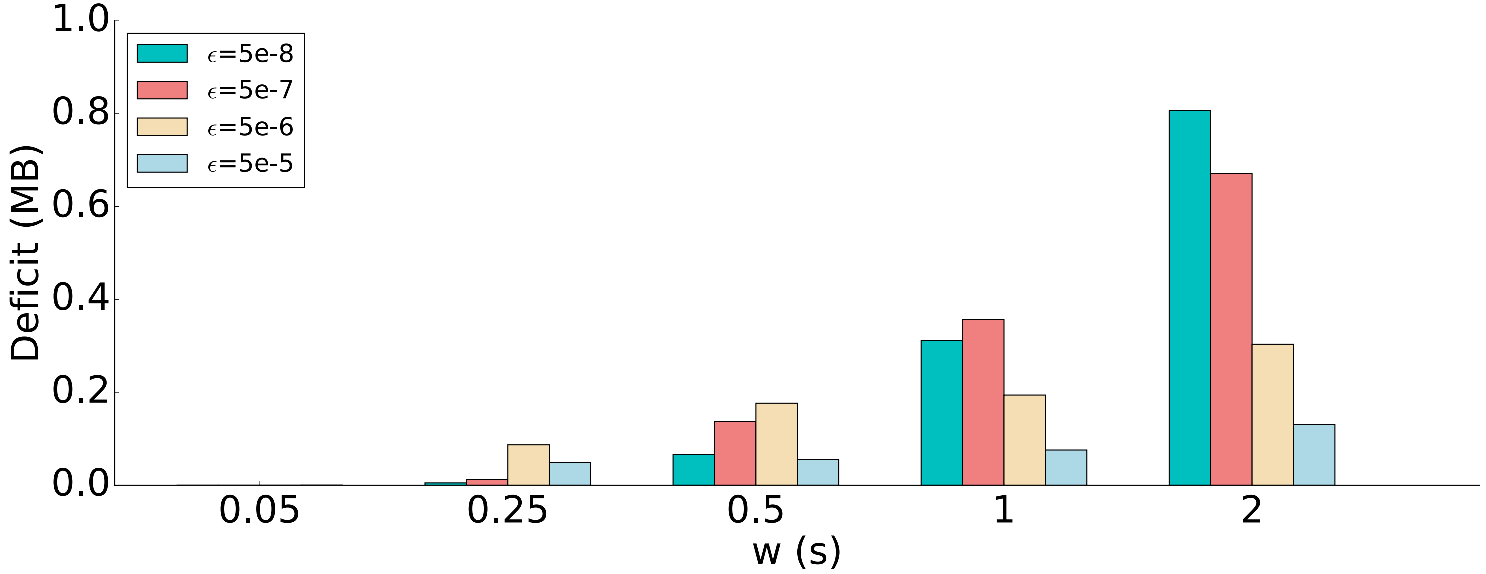


Utility Evaluation --- Deficit

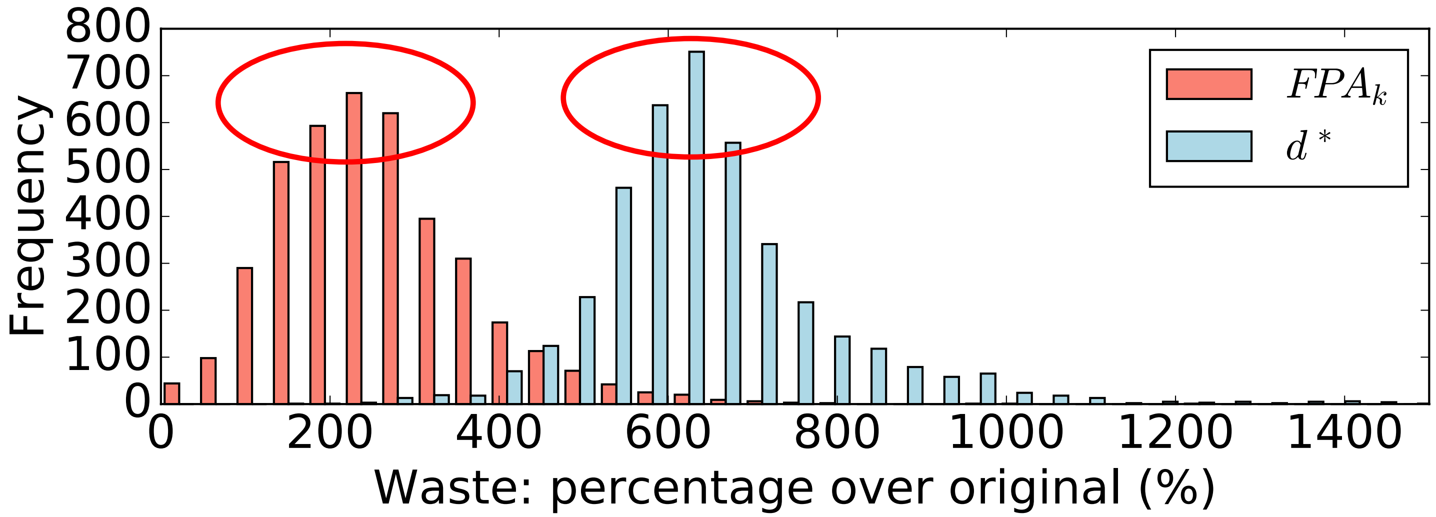
FPA_k

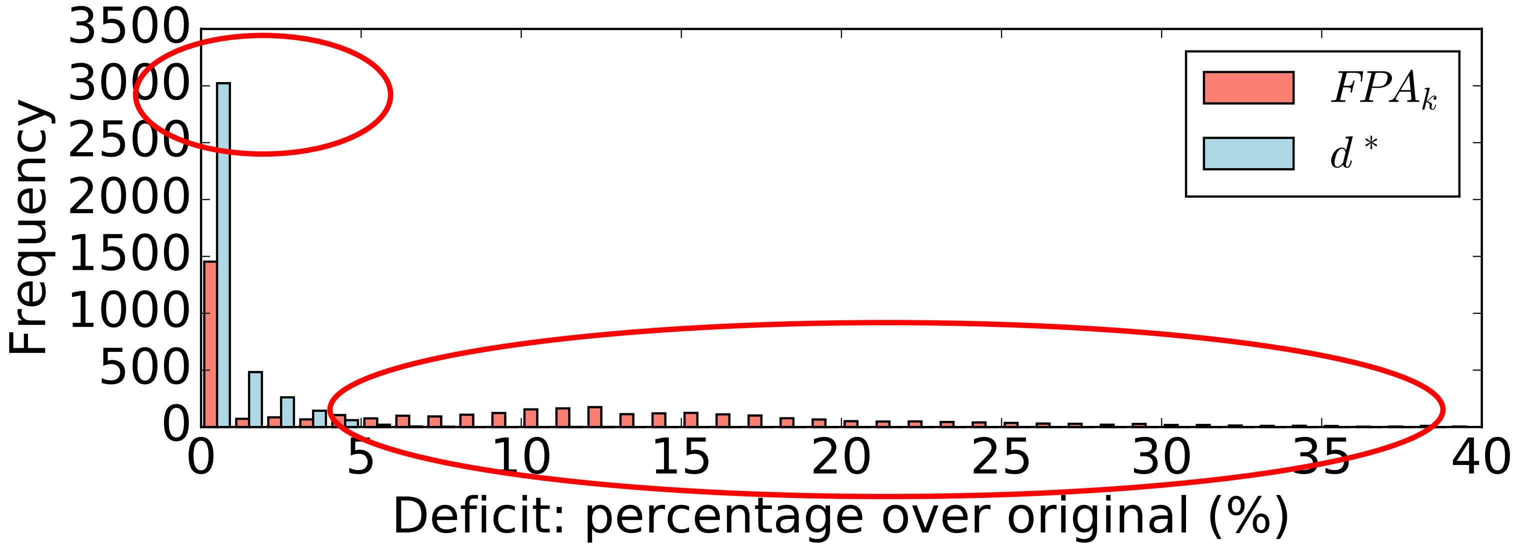


d^*



FPA_k vs. d*





Baseline Accuracy (2.5%)

Lowest Waste

$$FPA_k(w = 2s, \epsilon = 0.5)$$

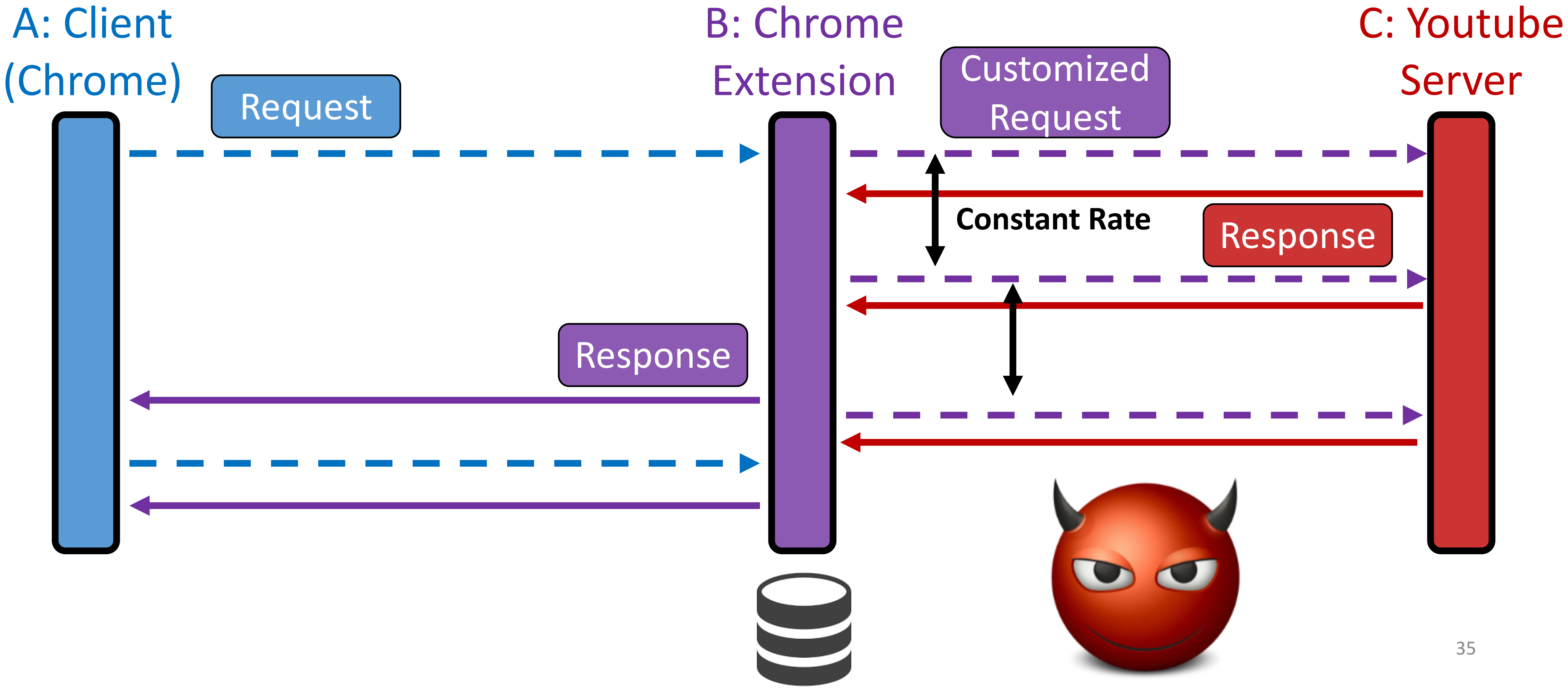
$$d^*(w = 0.5s, \epsilon = 5e - 6)$$

Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion

Implementation --- Workflow

- Chrome Extension: change the `range` in the HTTP request (FPA_k)



Implementation --- Effectiveness

- Dataset: 10 videos, 100 traces per video with extension
- 80% training, 20% test

- Settings: $FPA_k(w = 1s, \epsilon = 0.5) \quad w_A = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$

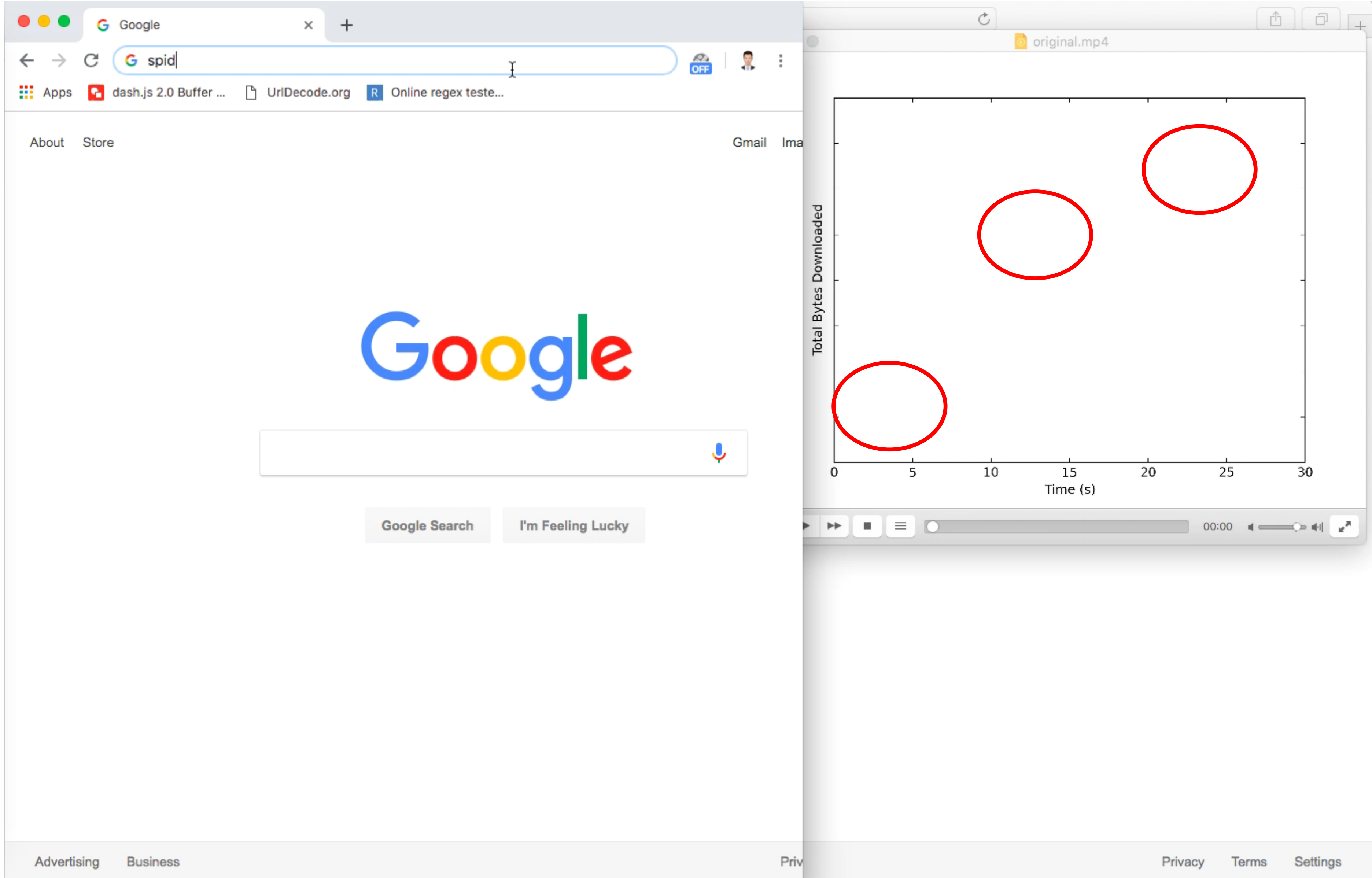
- Features:
  - up/down/total bytes per bin (BPB)
  - up/down/total packets per bin (PPB)
  - up/down/total average packet length per bin (LPB)
  - up/down/total bursts (BURST)
  - the combination of all 12 features (ALL)

Implementation --- Effectiveness

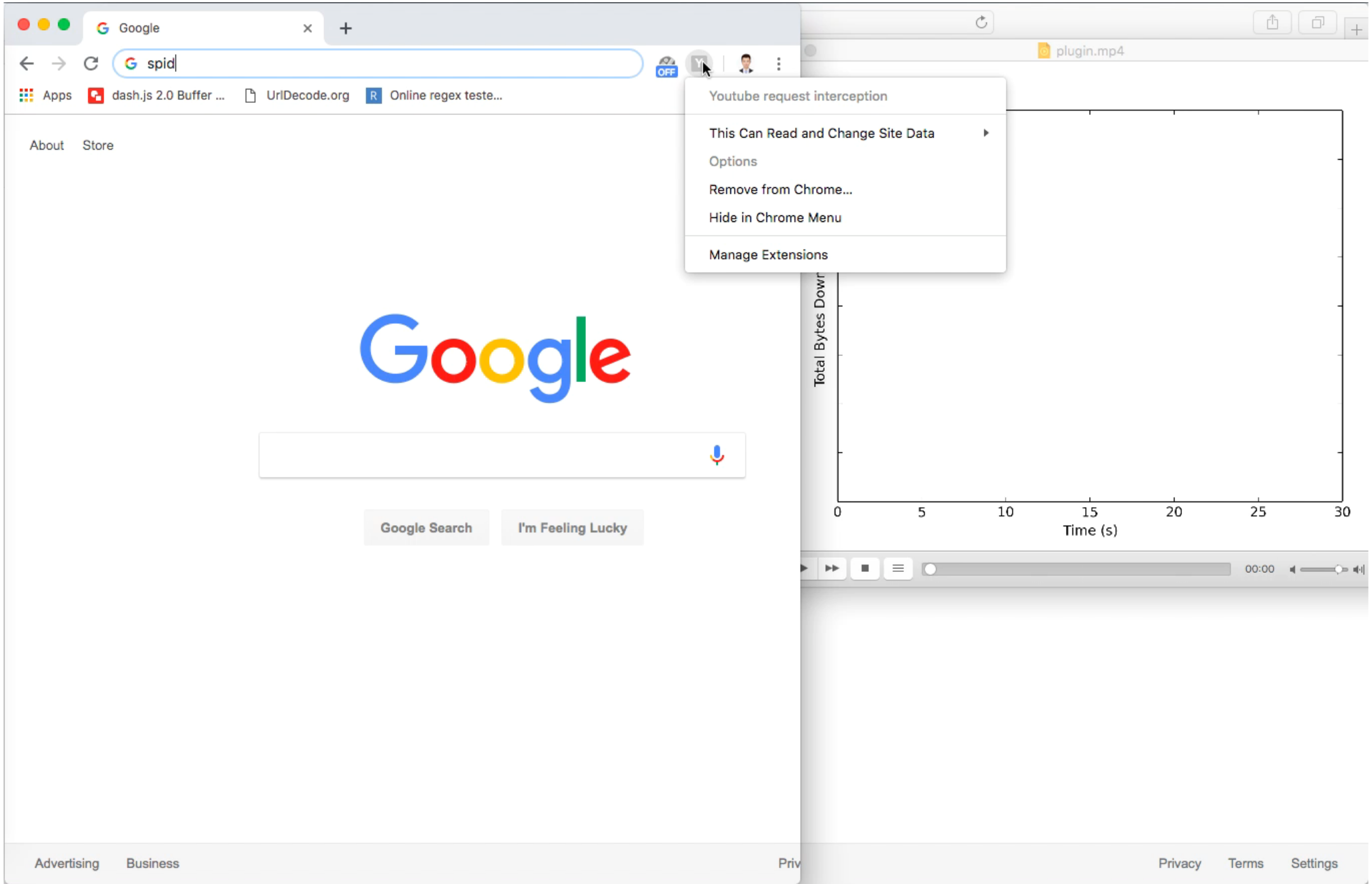
- Classification result (CNN)

$w_A(s)$	BPB_{up}	BPB_{down}	BPB	PPB_{up}	PPB_{down}	PPB	LPB_{up}	LPB_{down}	LPB	$BURST_{up}$	$BURST_{down}$	$BURST$	ALL
0.05	0.16	0.12	0.16	0.12	0.16	0.14	0.14	0.13	0.16	0.14	0.15	**0.16**	0.13
0.25	0.20	0.16	0.22	0.18	0.16	0.20	0.12	0.08	0.16	**0.23**	0.14	0.19	0.21
0.5	0.19	0.12	**0.22**	0.14	0.16	0.20	0.14	0.08	0.10	0.19	0.14	0.15	0.20
1	0.16	0.14	0.18	0.14	**0.19**	0.13	0.10	0.10	0.11	0.16	0.14	0.12	0.18
2	0.14	0.12	0.16	0.13	0.14	0.16	0.10	0.10	0.09	0.16	0.16	**0.19**	0.17

Implementation --- Demo: original



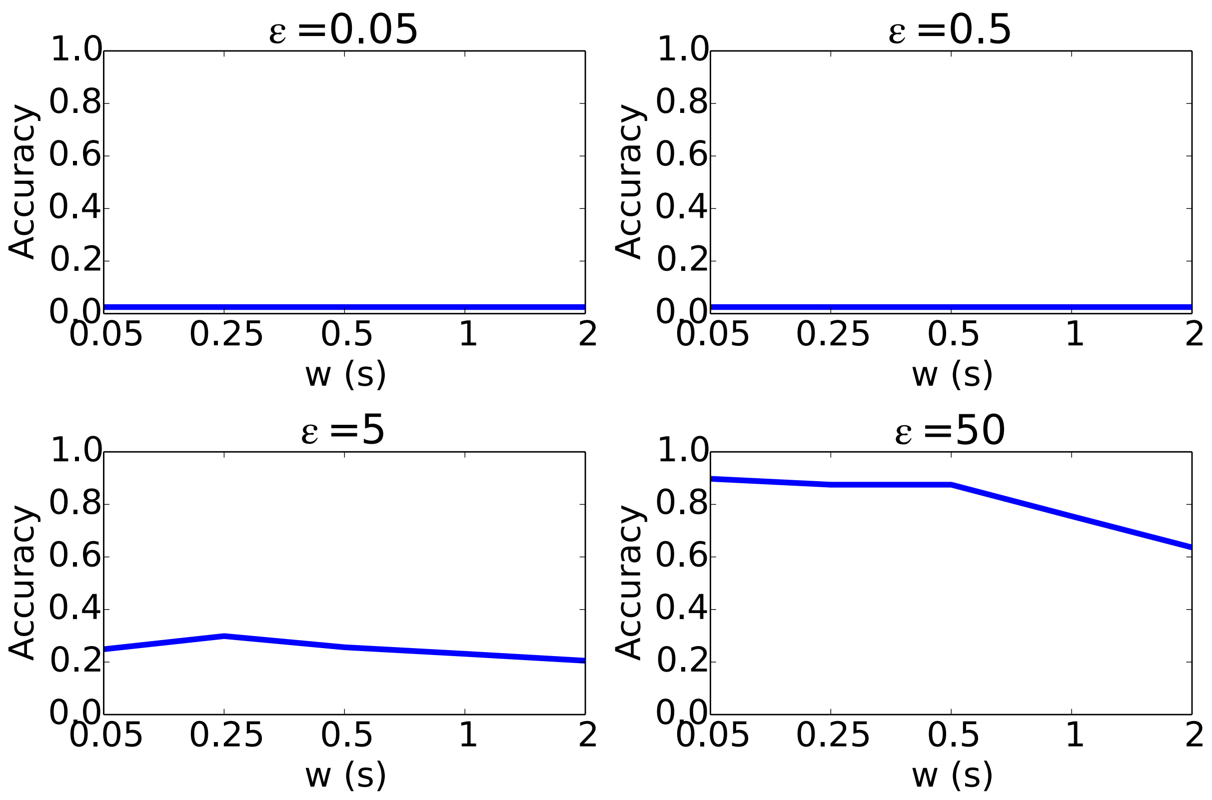
Implementation --- Demo: w. extension

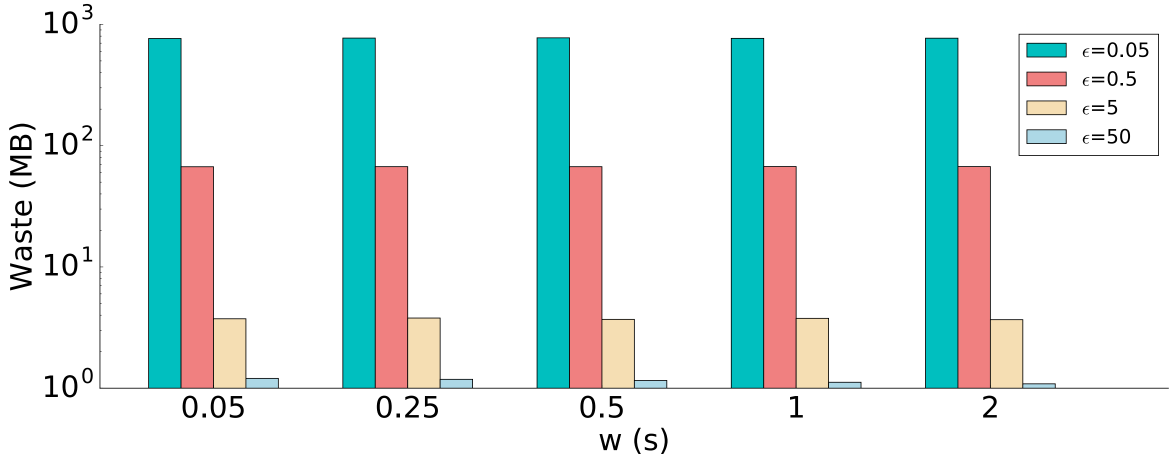


Outline

Discussion

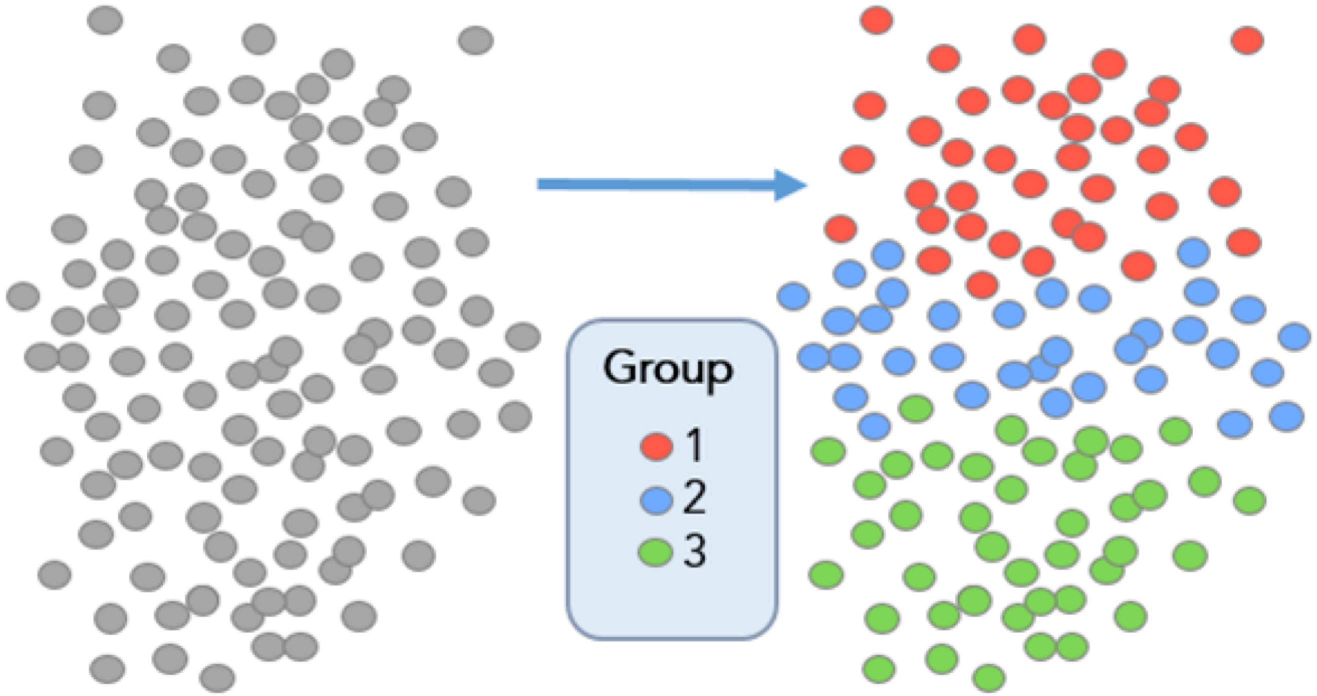
- Reducing waste:
  - Lowering clip bound (e.g. [0, 1GB] -> [0, 100MB])
  - Increasing ϵ





Discussion

- Leakage through video length
  - Cannot prevent due to utility loss
  - Possible solution: grouping the videos by length and padding them to the longest length in each group



Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion

Conclusion

- We borrowed techniques from adversarial ML and differential privacy to address privacy concerns of streaming traffic
- We showed that differential privacy effectively defeats inference-based traffic analysis, while remains agnostic to the ML classifiers
- Results suggested that the two differentially private mechanisms offer good security protection with moderate utility loss

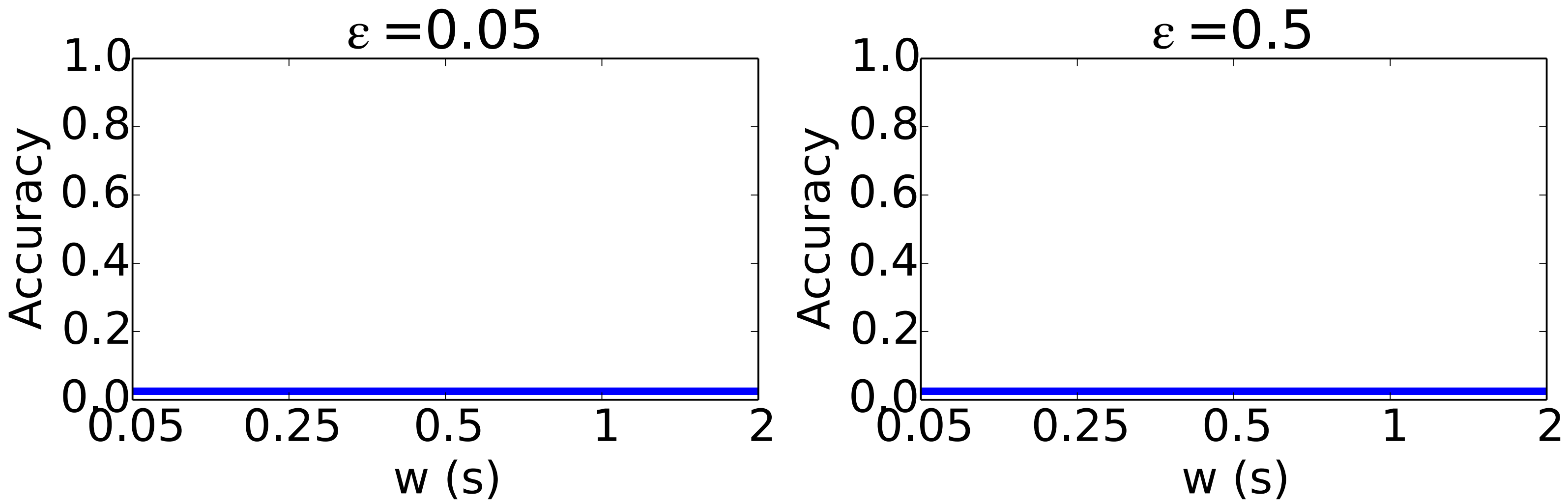
Thanks for listening!

Xiaokuan Zhang
zhang.5840@osu.edu

Backup Slides

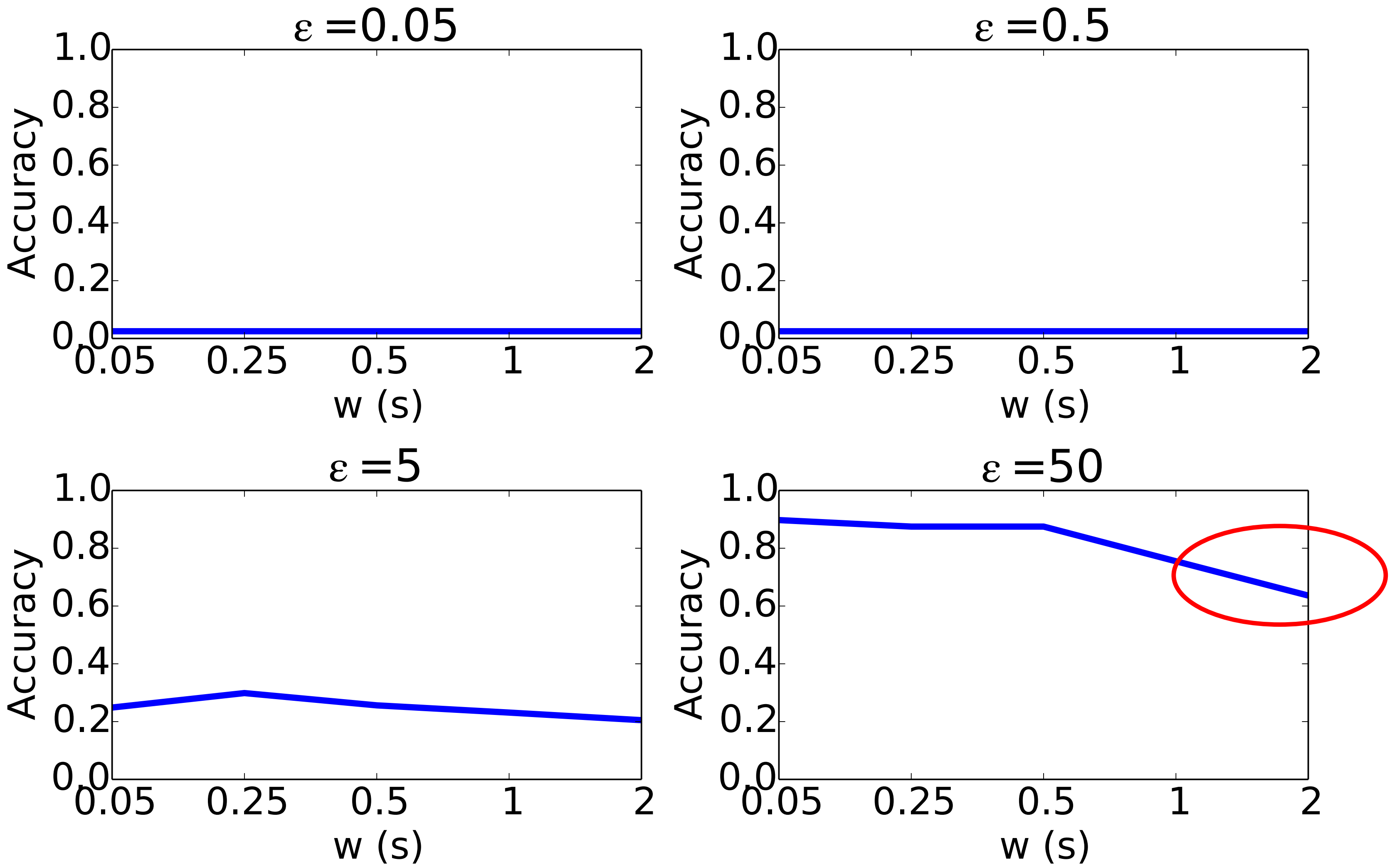
Security Evaluation --- FPA_k

$$w_A = w\text{: effect of } w$$



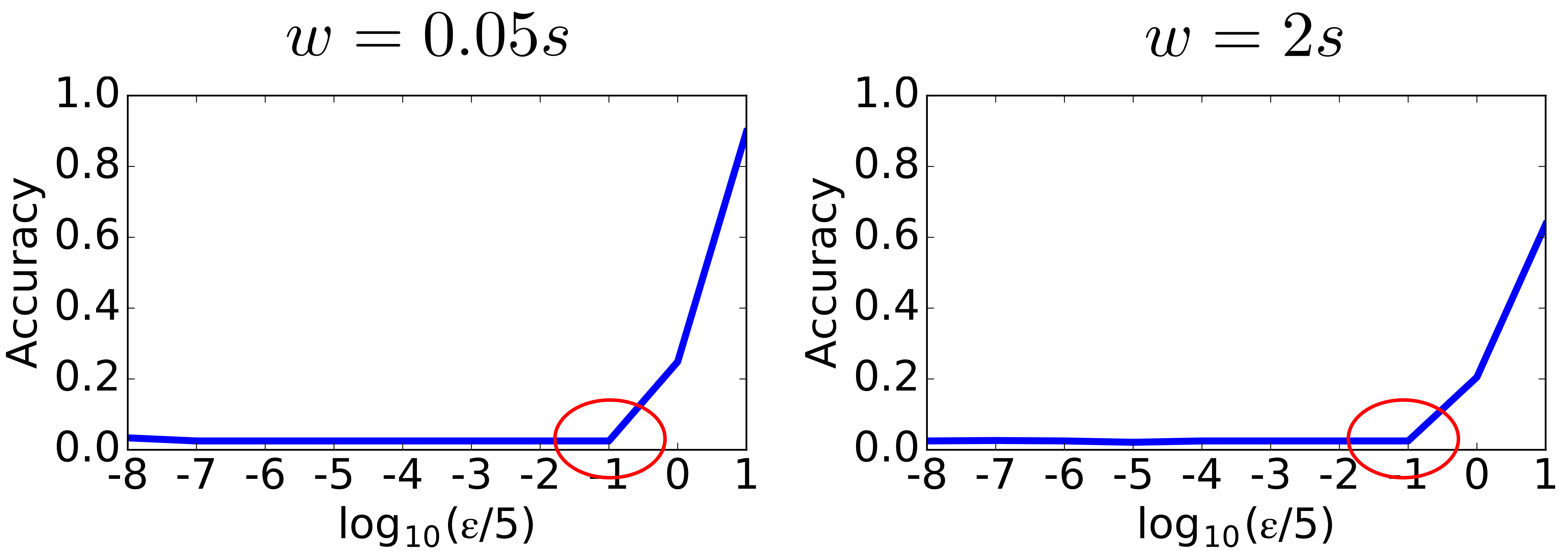
Security Evaluation --- FPA_k

$$w_A = w\text{: effect of } w$$



Security Evaluation --- FPA_k

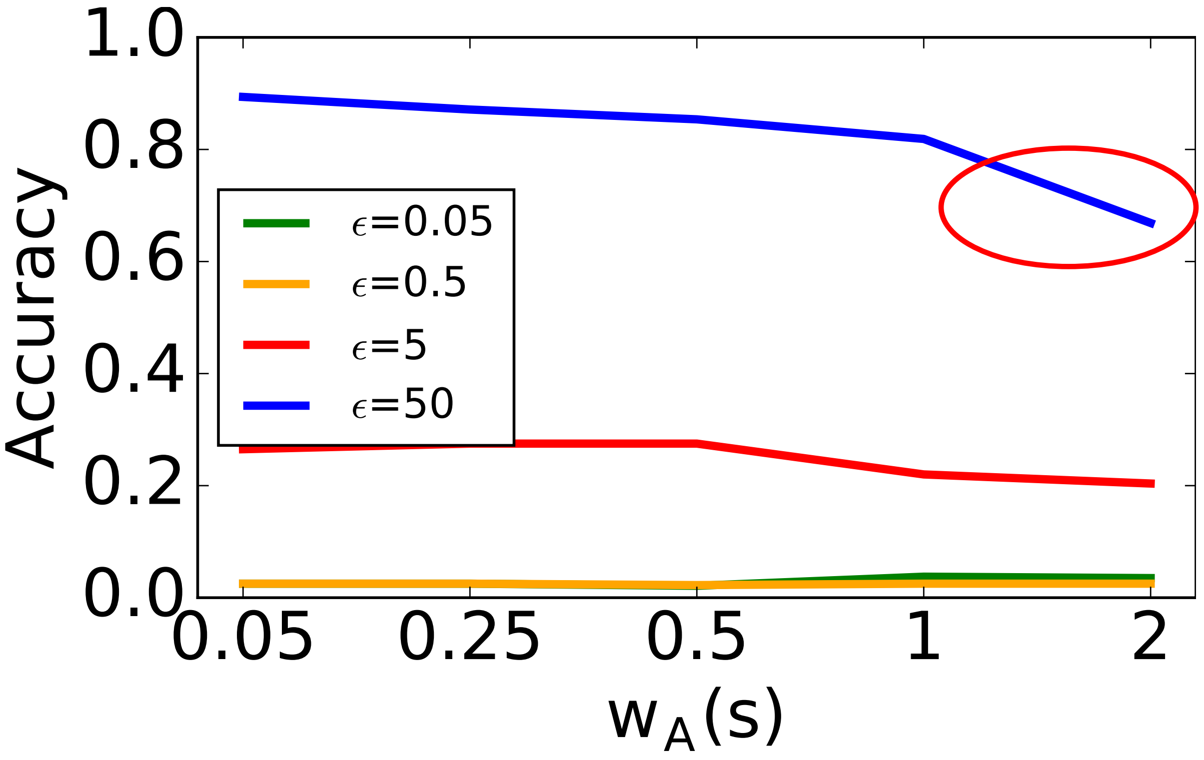
$$w_A = w\text{: effect of } \epsilon$$



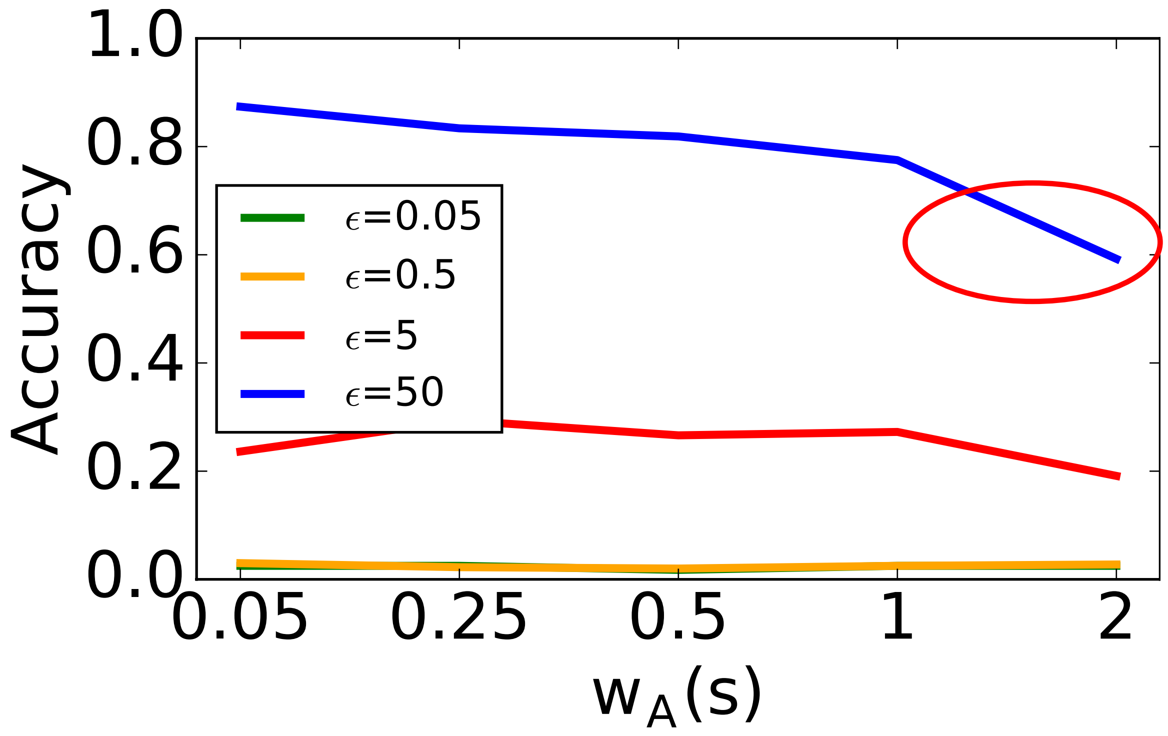
Security Evaluation --- FPA_k

$$w \neq w_A$$

$$w = 0.05s$$

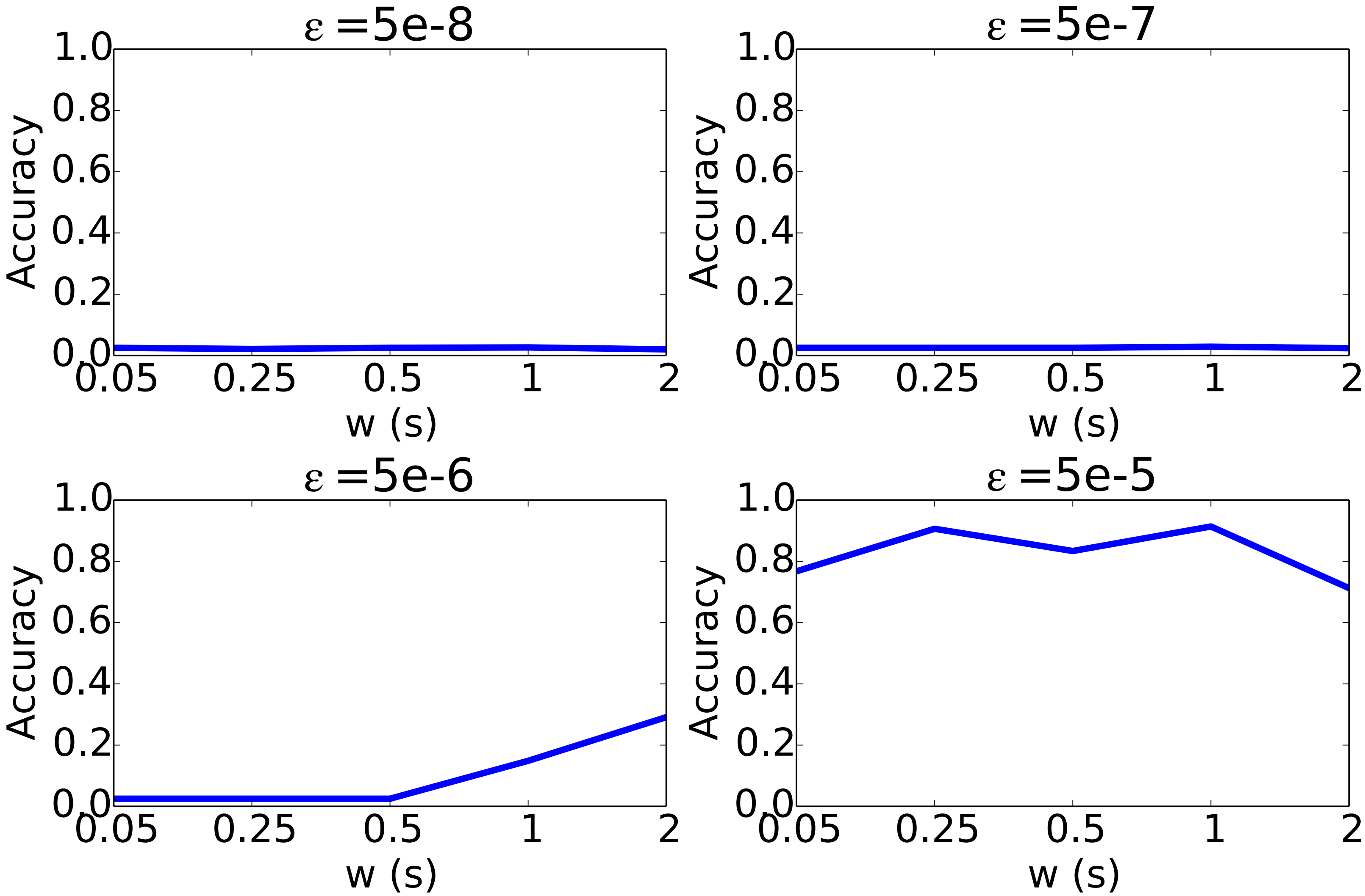


$$w = 2s$$



Security Evaluation

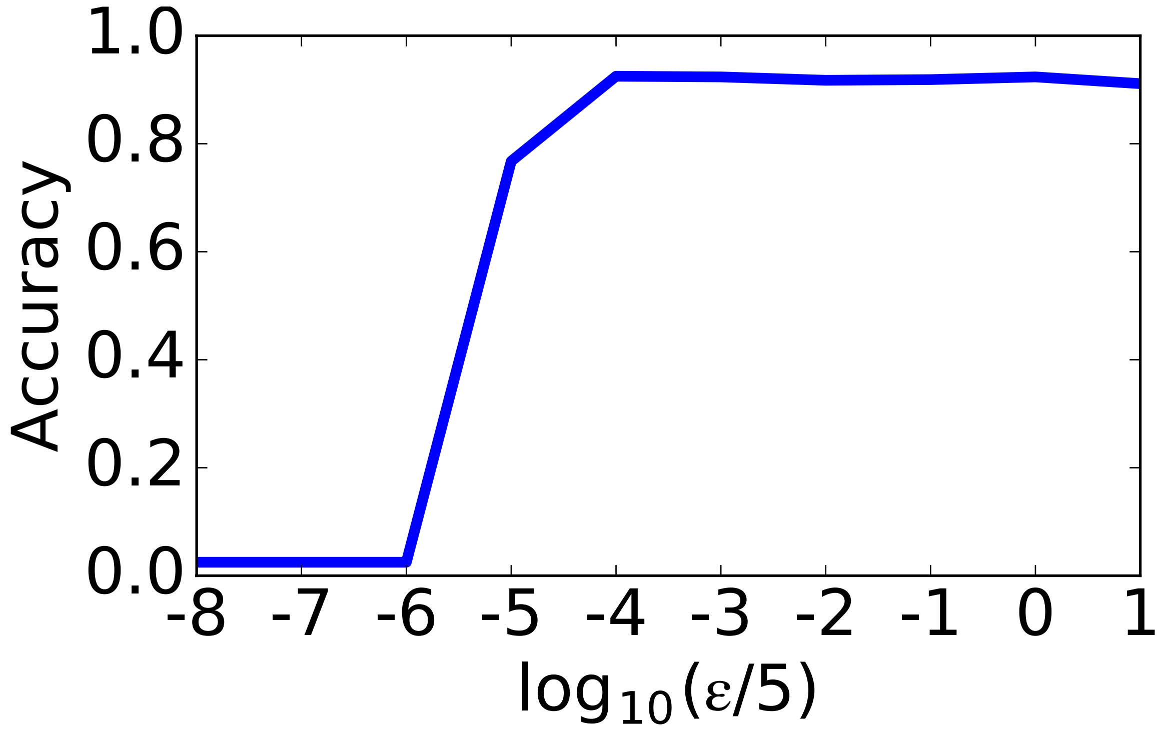
$$w_A = w\text{: effect of } w$$ d*



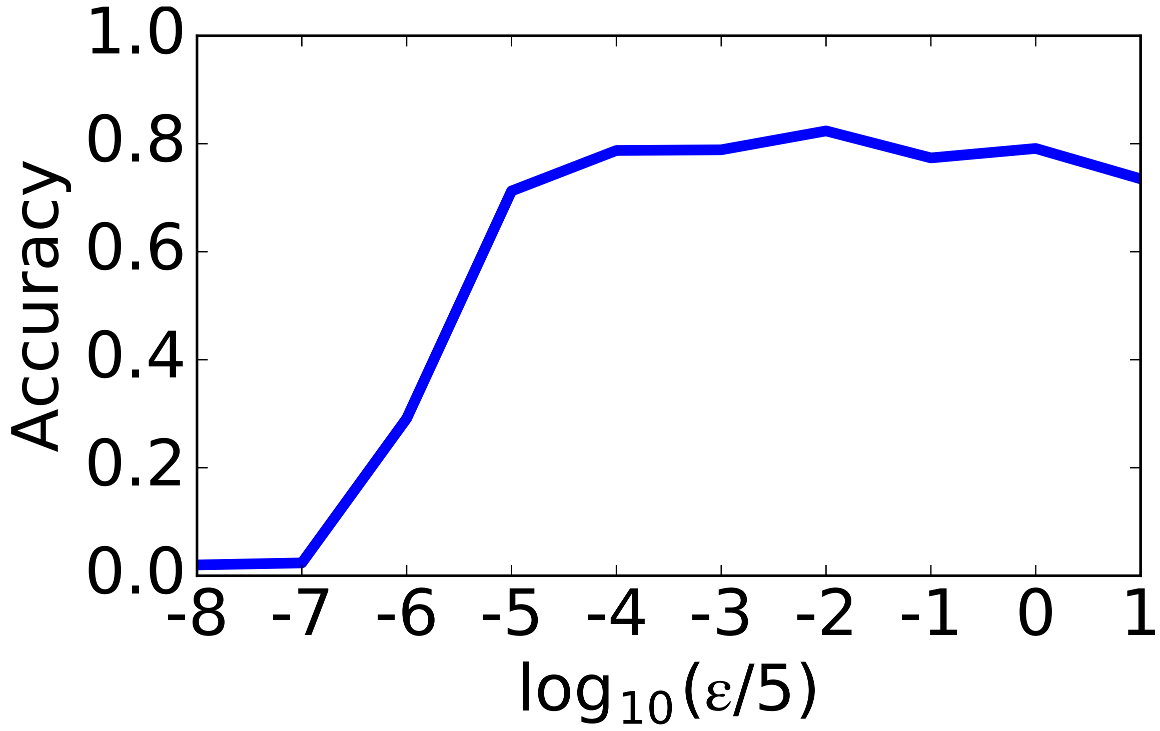
Security Evaluation

$$w_A = w\text{: effect of } \epsilon \;\; \textcolor{red}{\mathtt{d*}}$$

$$w = 0.05s$$

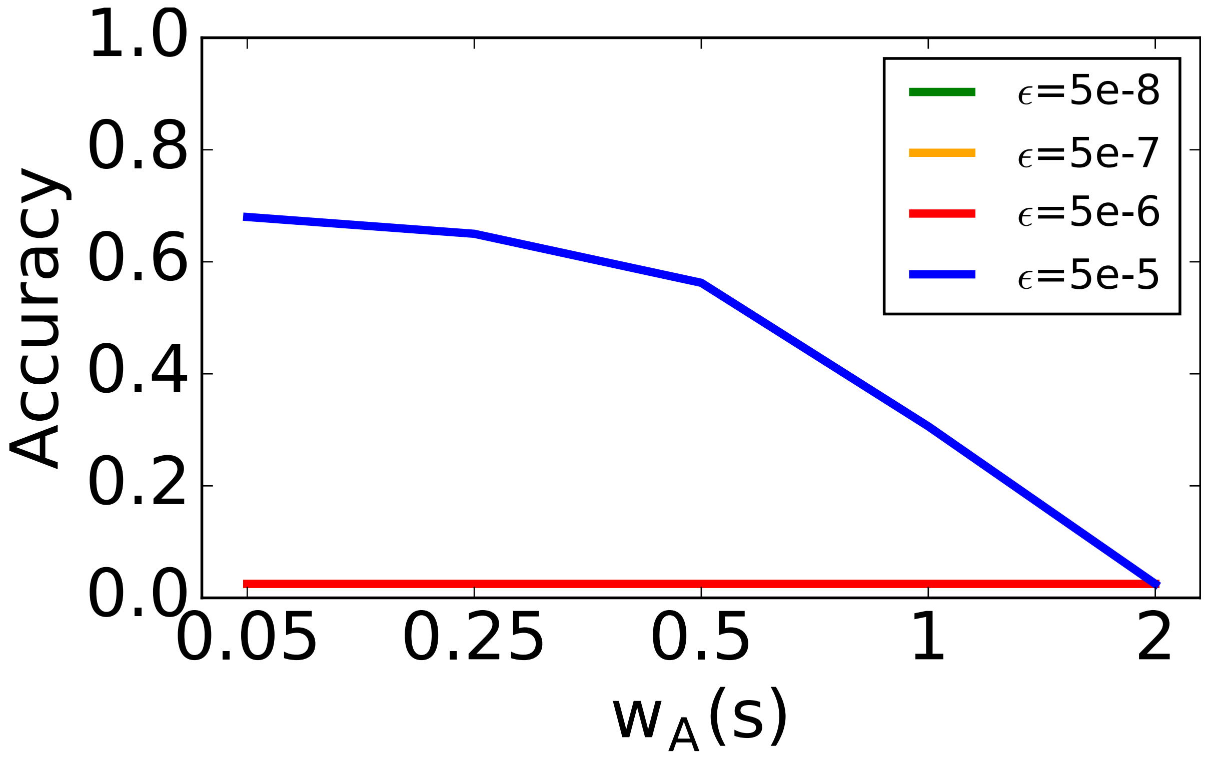


$$w = 2s$$

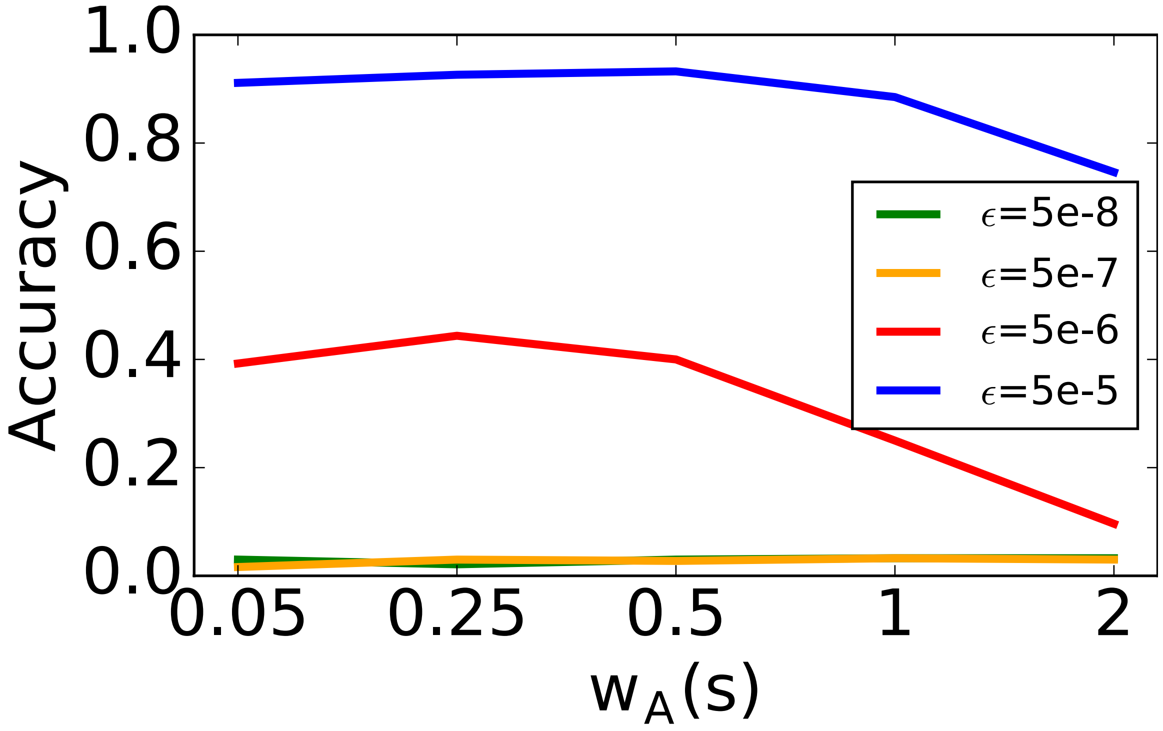


Security Evaluation

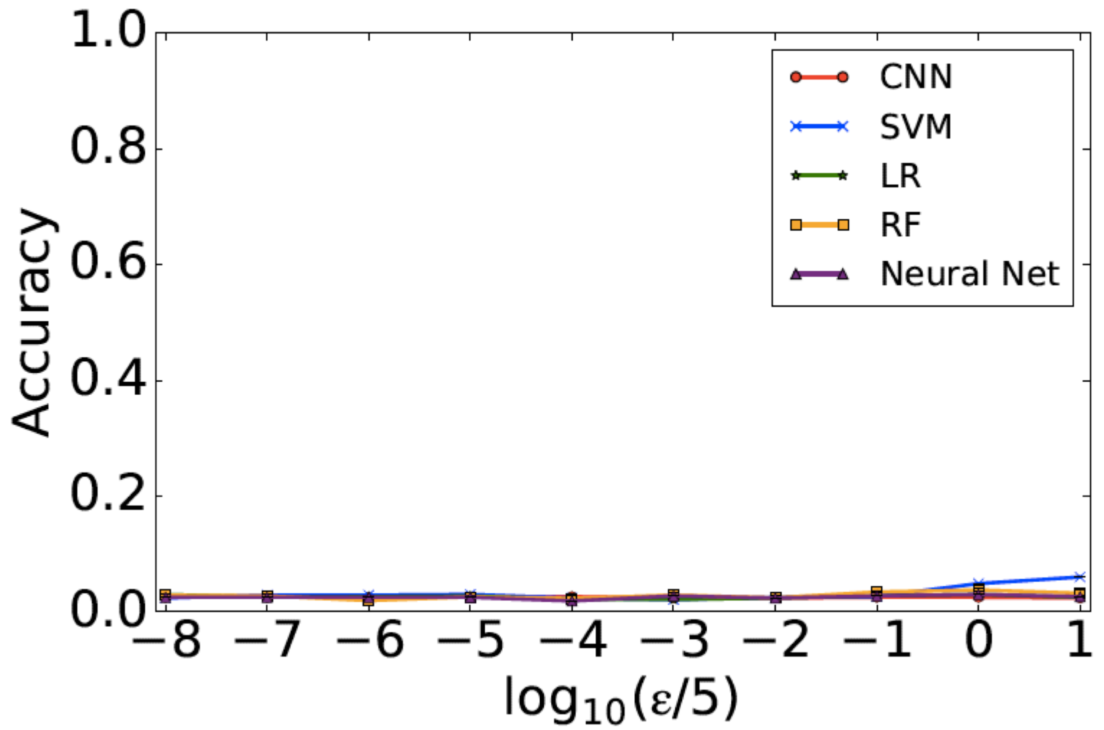
d*: w = 0.05s



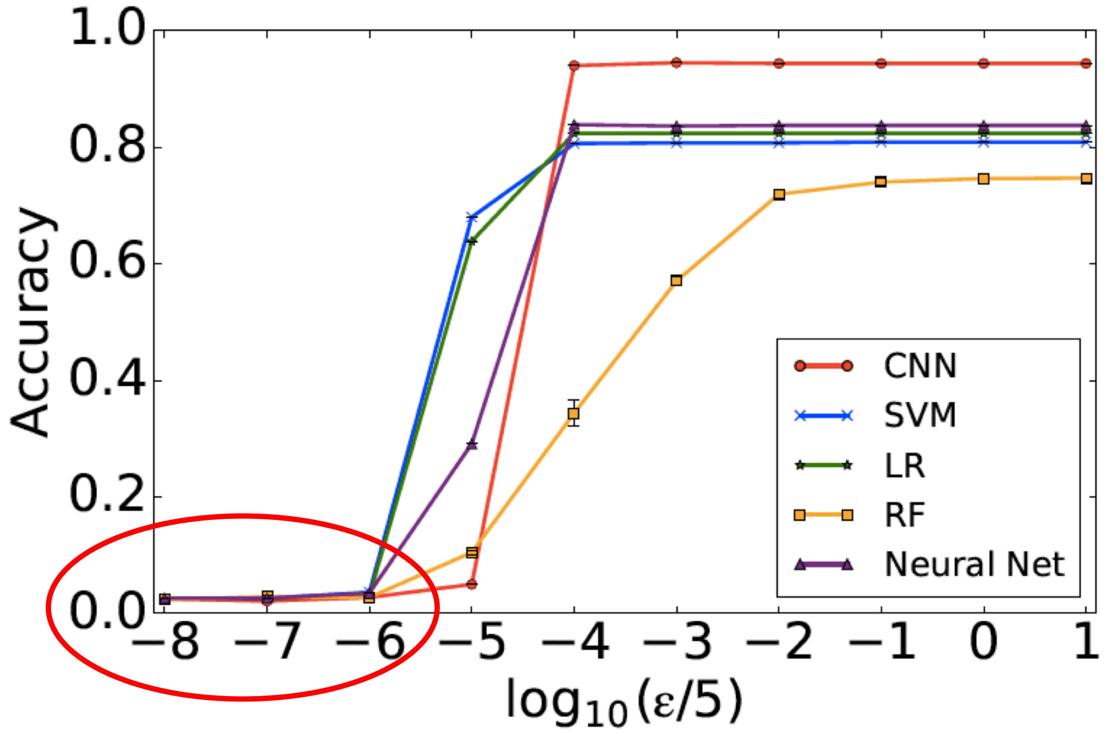
d*: w = 2s



Security Evaluation --- Train w. clean, test w. noised



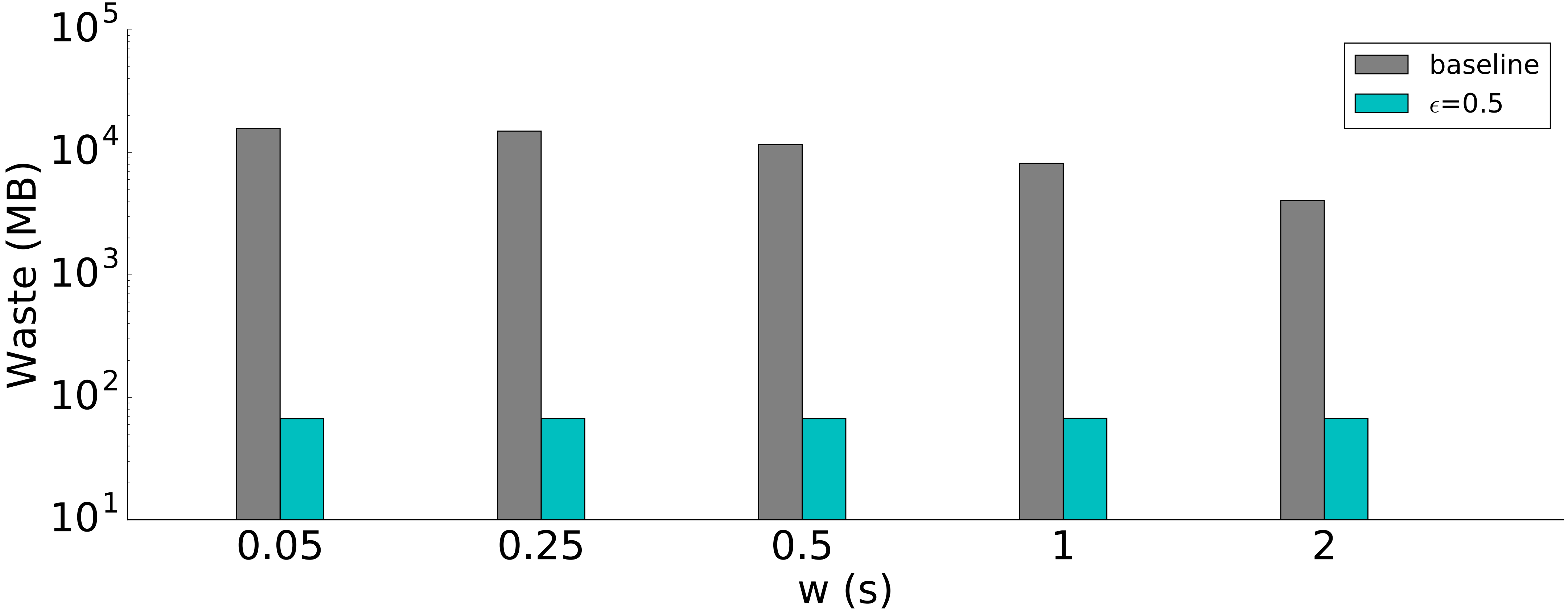
(a) FPA_k



(b) d^*

Baseline Approach

- Window size: w seconds
- Max value of all bins of all videos (4000 traces): C
- Baseline defense mechanism: C bytes per w seconds (all videos)



Optimal Attacker

- The Attacker has the knowledge of distribution of both clean data and noised data (but not the mapping between the two)

- First try to remove noise, then perform classification

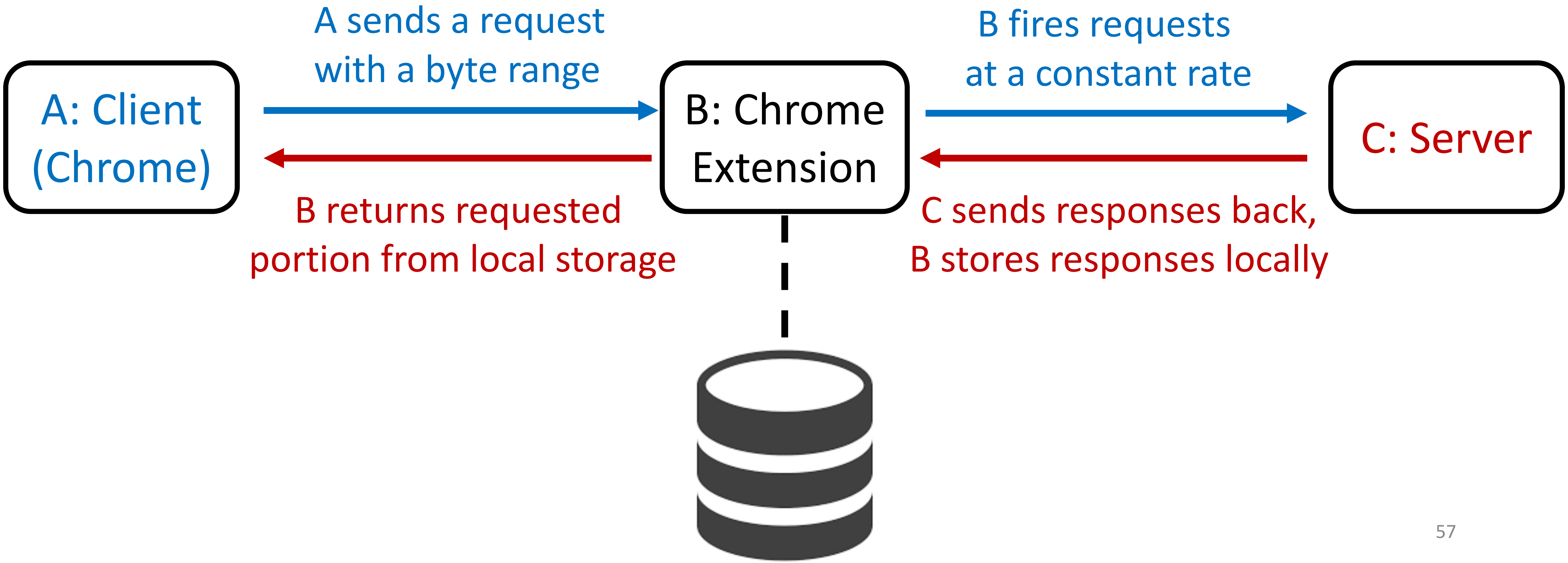
	FPA_k				d^*			
$w(s)$ \ ϵ	0.05	0.5	5	50	5e-8	5e-7	5e-6	5e-5
0.05	0.03	0.03	0.25	0.89	0.03	0.03	0.03	0.72
0.25	0.03	0.03	0.30	0.89	0.03	0.03	0.03	0.89
0.5	0.03	0.03	0.27	0.87	0.02	0.02	0.03	0.86
1	0.03	0.03	0.27	0.80	0.02	0.02	0.11	0.89
2	0.03	0.03	0.17	0.65	0.03	0.03	0.10	0.75

Improvement of accuracy: $\leq 2\%$

Naldi et al. "Differential privacy for counting queries: can bayes estimation help uncover the true value?", arXiv:1407.0116.

Implementation --- Workflow

- Chrome Extension: change the byte range in the HTTP request



Discussion

- Comparing FPA_k with d*
  - Accuracy ←→ Security Guarantee
  - FPA_k requires the knowledge of the entire time series

$$Accuracy \neq$$