# Burst ORAM: Minimizing ORAM Response Times for Bursty Access Patterns

**Jonathan Dautrich,** Emil Stefanov, Elaine Shi

Usenix Security, San Diego

August 22, 2014

UCR
UNIVERSITY OF CALIFORNIA, RIVERSIDE

# Problems to Solve

- Access pattern privacy for outsourced data

**Oblivious RAM**

- Practical response times for bursty workloads

**3 Hours** ⇒ **56 ms**

ObliviStore ORAM — Burst ORAM

99.9% response times, NetApp trace simulation, 50ms latency
32TB capacity, 100GB client storage, 400Mbps bandwidth

# Outline

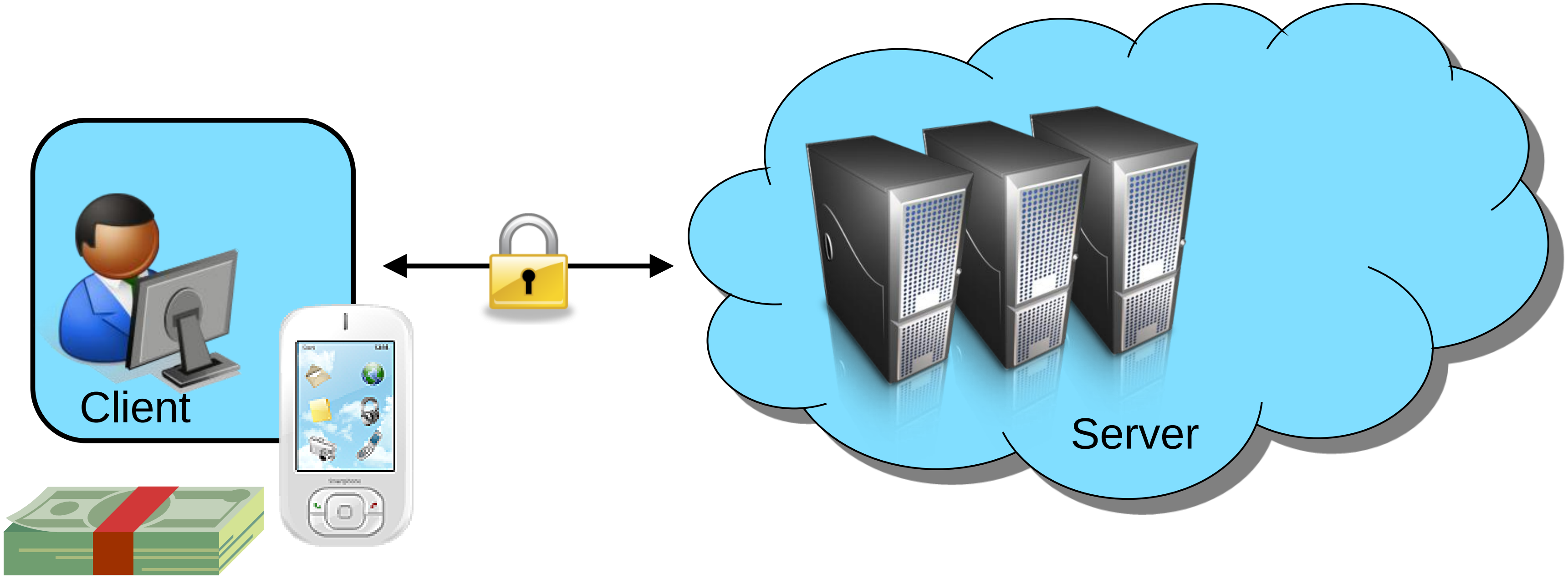
- Private Data Outsourcing
  - Oblivious RAM
- Practical ORAM Response Times
  - Burst ORAM
- Burst ORAM Details
- Results

# Outline

- Private Data Outsourcing
  - Oblivious RAM
- Practical ORAM Response Times
  - Burst ORAM
- Burst ORAM Details
- Results

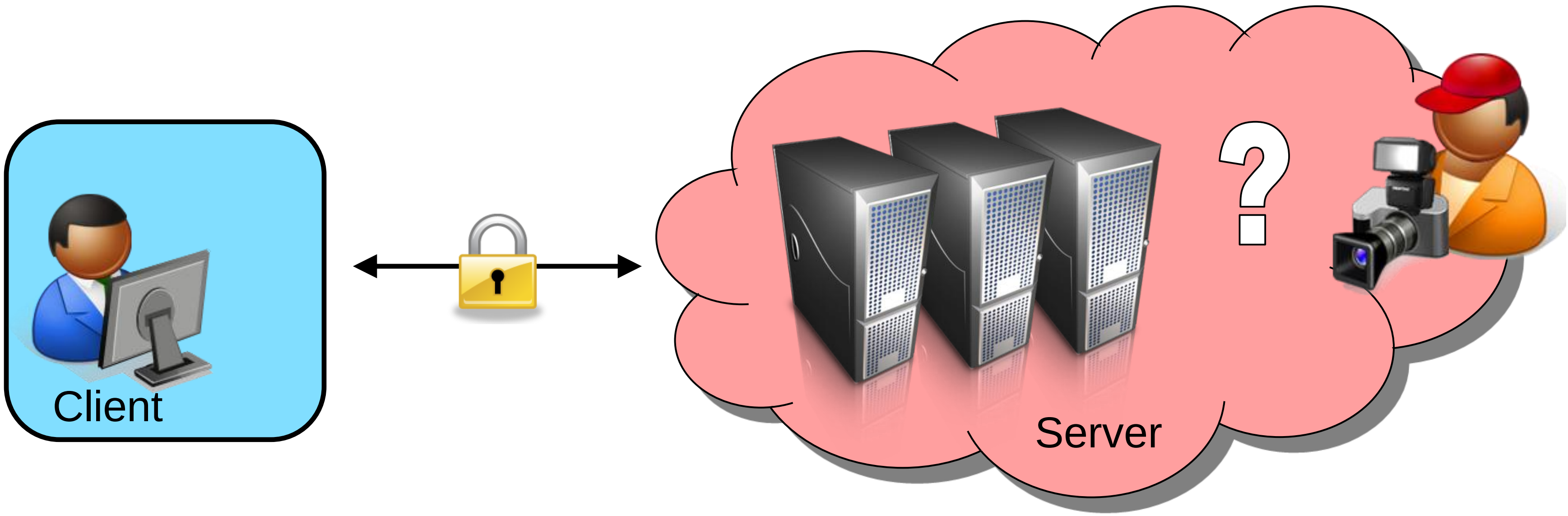
# Data Outsourcing (Cloud)

- Advantages
  - Accessibility and availability
  - IT overhead savings



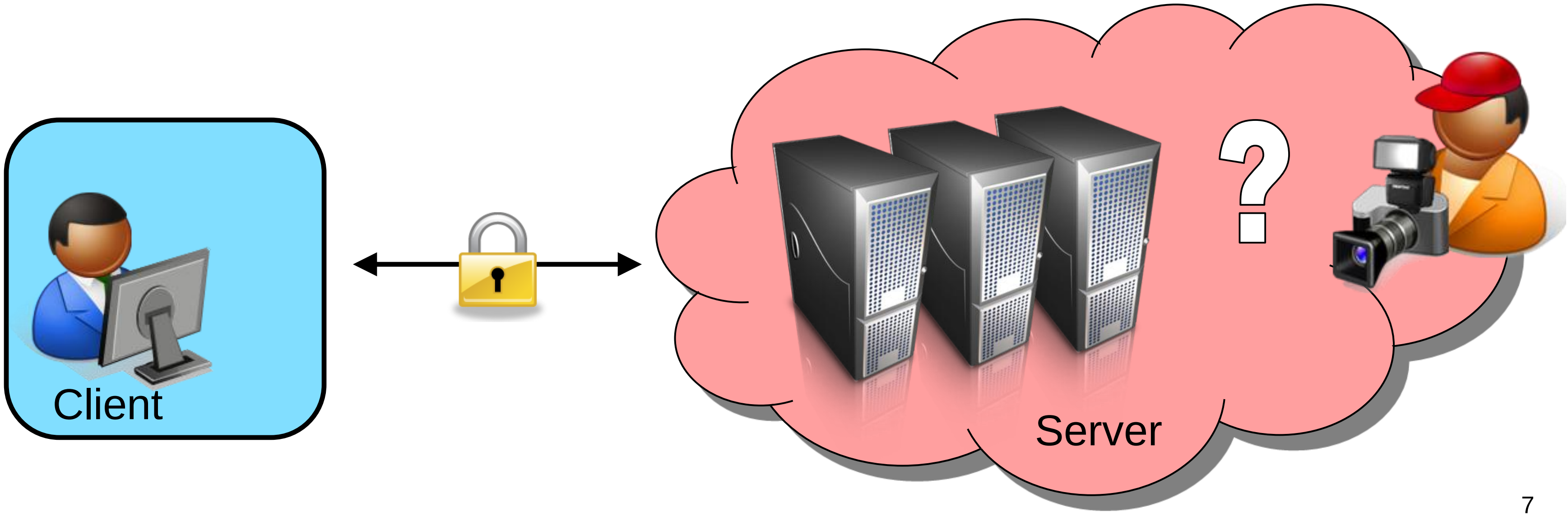
# Why care about data privacy?

- Inexpensive but untrusted provider
- Privacy Regulations
  - FERPA, HIPAA, ITAR, etc.



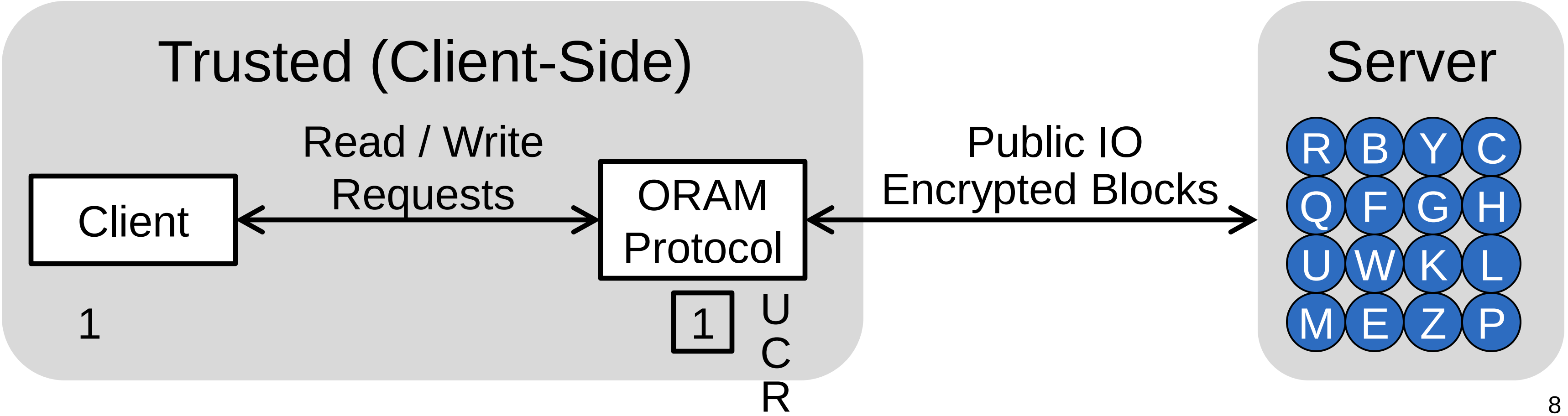
# Access Pattern Privacy

- Encryption alone is insufficient
- *Access patterns* leak information
  - Patterns in plaintext (Dautrich & Ravishankar, EDBT 2013)
  - Search query contents (Islam et al., NDSS 2012)



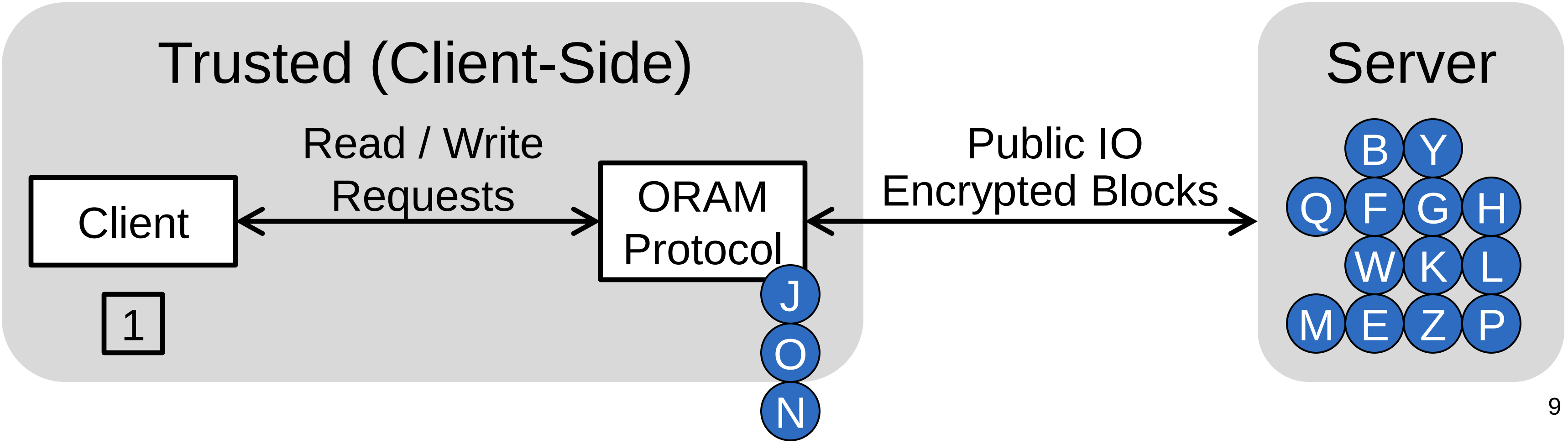
# Oblivious RAM (Goldreich & Ostrovsky, 1996)

- Provable access pattern privacy
- ORAM translates client requests to public IO
  - Online IO: Needed to satisfy request



# Oblivious RAM (Goldreich & Ostrovsky, 1996)

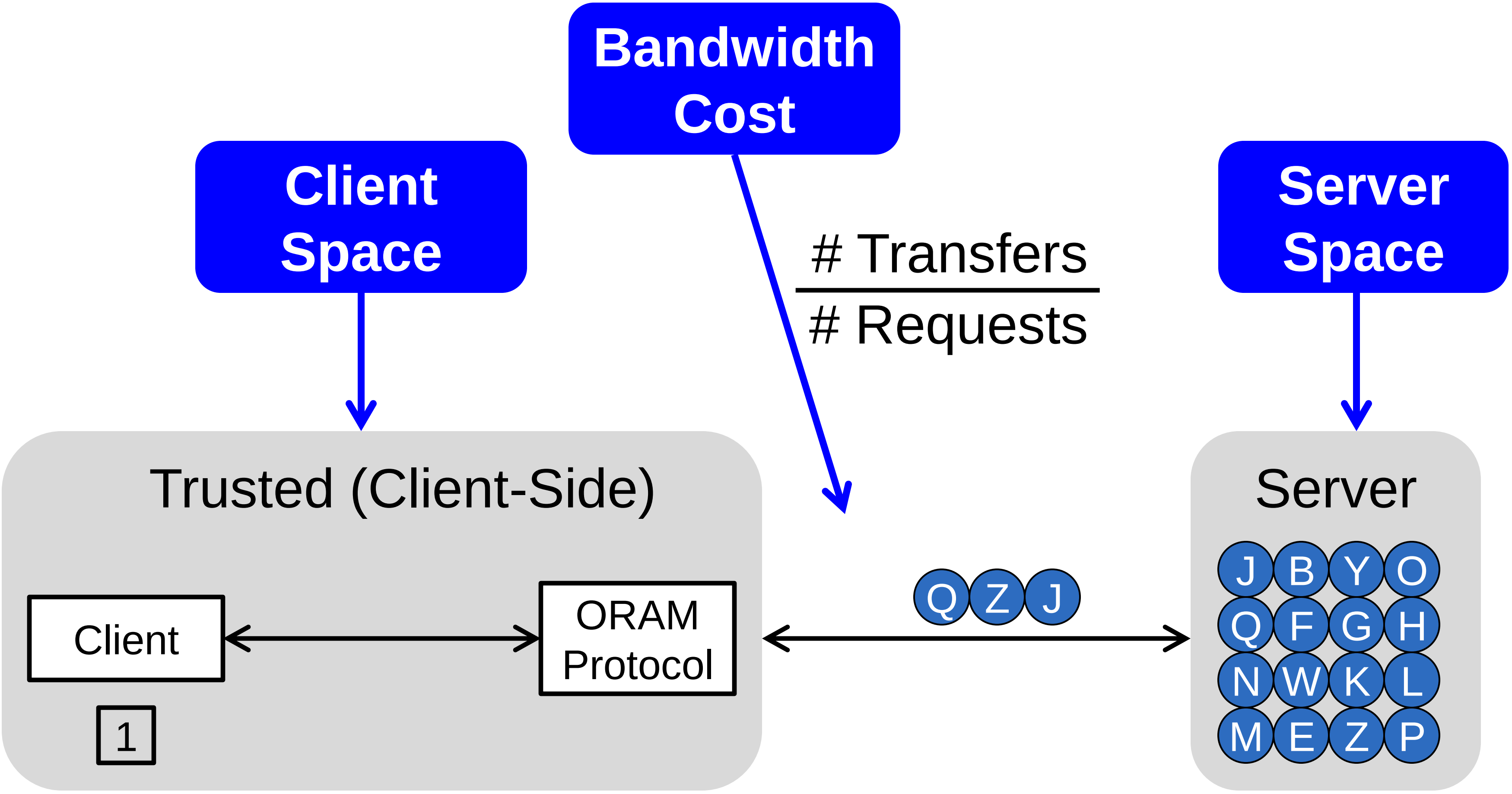
- Provable access pattern privacy
- ORAM translates client requests to public IO
  - Online IO: Needed to satisfy request
  - Offline IO: After request completes (shuffling)



# Outline

- Private Data Outsourcing
  - Oblivious RAM
- Practical ORAM Response Times
  - Burst ORAM
- Burst ORAM Details
- Results

# ORAM Costs

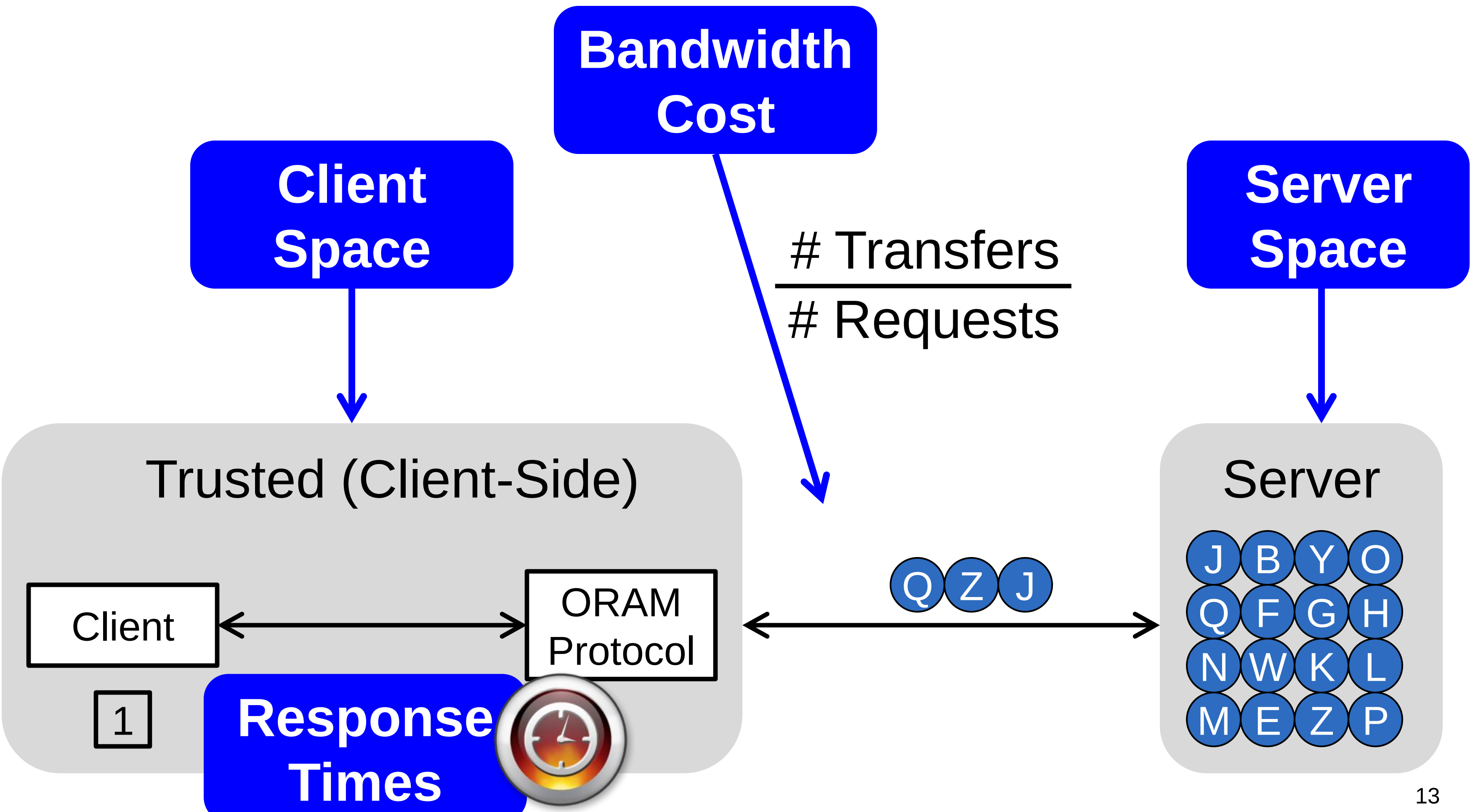

Bandwidth Cost
Client Space
Server Space
# Transfers
# Requests
Trusted (Client-Side)
Client
ORAM Protocol
1
Q Z J
Server
J B Y O
Q F G H
N W K L
M E Z P

# Some Prior ORAM Schemes

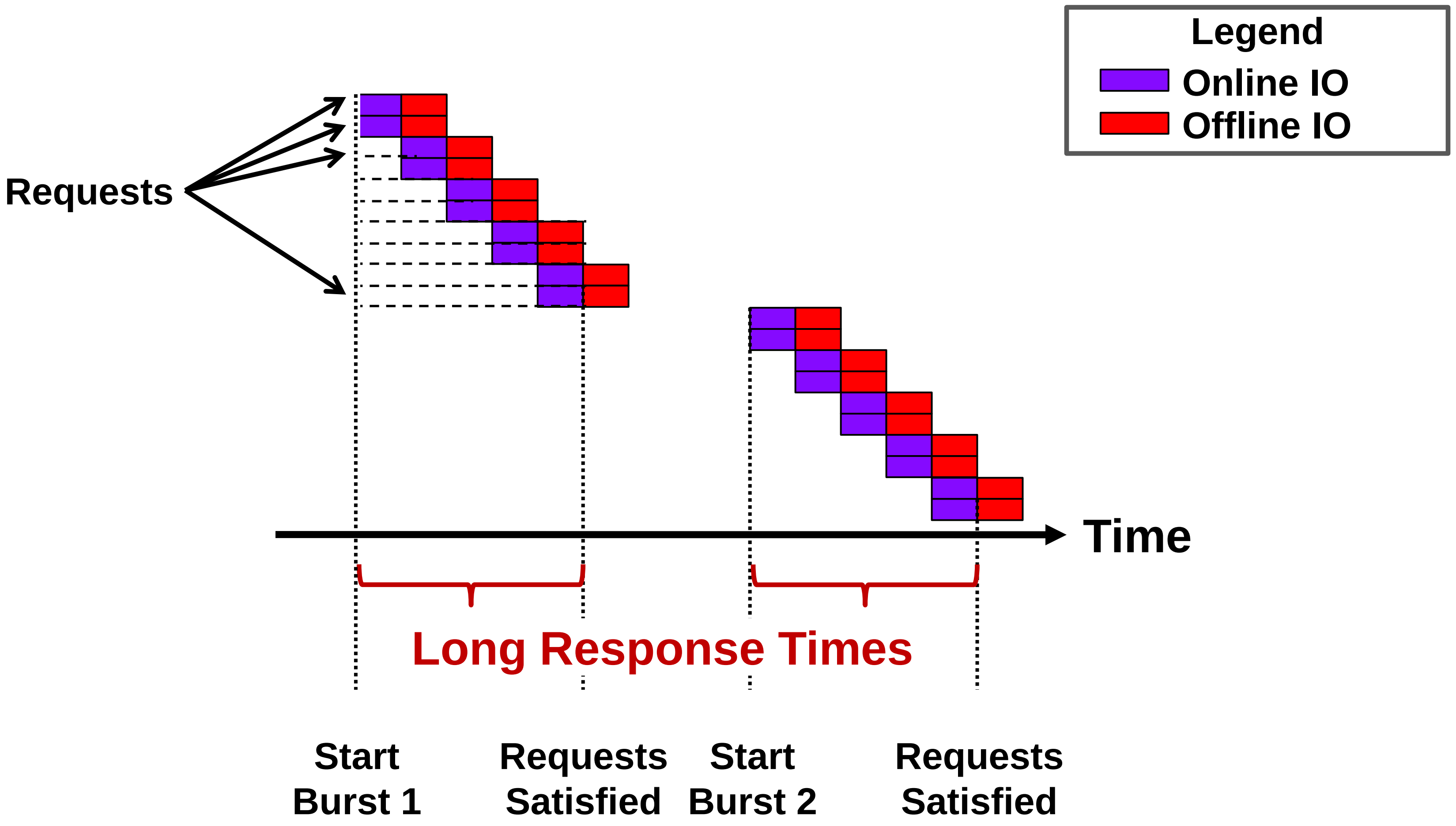
*N* blocks of size *B* (*B* generally 1KB to 1MB)

| Scheme | Client Space | Server Space | Bandwidth Cost |
|---|---|---|---|
| Goldreich & Ostrovsky 1996 | O(B log N) | O(BN log N) | $O(\log^3 N)$ |
| Kushilevitz et al. 2012 | O(B) | O(BN) | $O(\log^2 N / \log \log N)$ |
| Goodrich et al. 2012 | $O(BN^{1/c})$ | O(BN) | O(log N) |
| Stefanov et al. 2013 (ObliviStore) | $\sim N \log_2 N + BN^{1/2}$ | ~2BN-4BN | $\sim \log_2 N$ |

# ORAM Costs


Bandwidth Cost
Client Space
Server Space
# Transfers
# Requests
Trusted (Client-Side)
Server
Client
ORAM Protocol
1
Response Times
Q Z J
J B Y O
Q F G H
N W K L
M E Z P

# Bursty Workload: Existing ORAMs



# Burst ORAM

- Goals
  - Minimize burst response times
  - Keep total bandwidth cost low
- Based on ObliviStore
  - Bandwidth-efficient
  - Large client space

# Burst ORAM Strategies

## 1: Reduce Online IO

- XOR technique

## 2: Delay Shuffling (Offline IO)

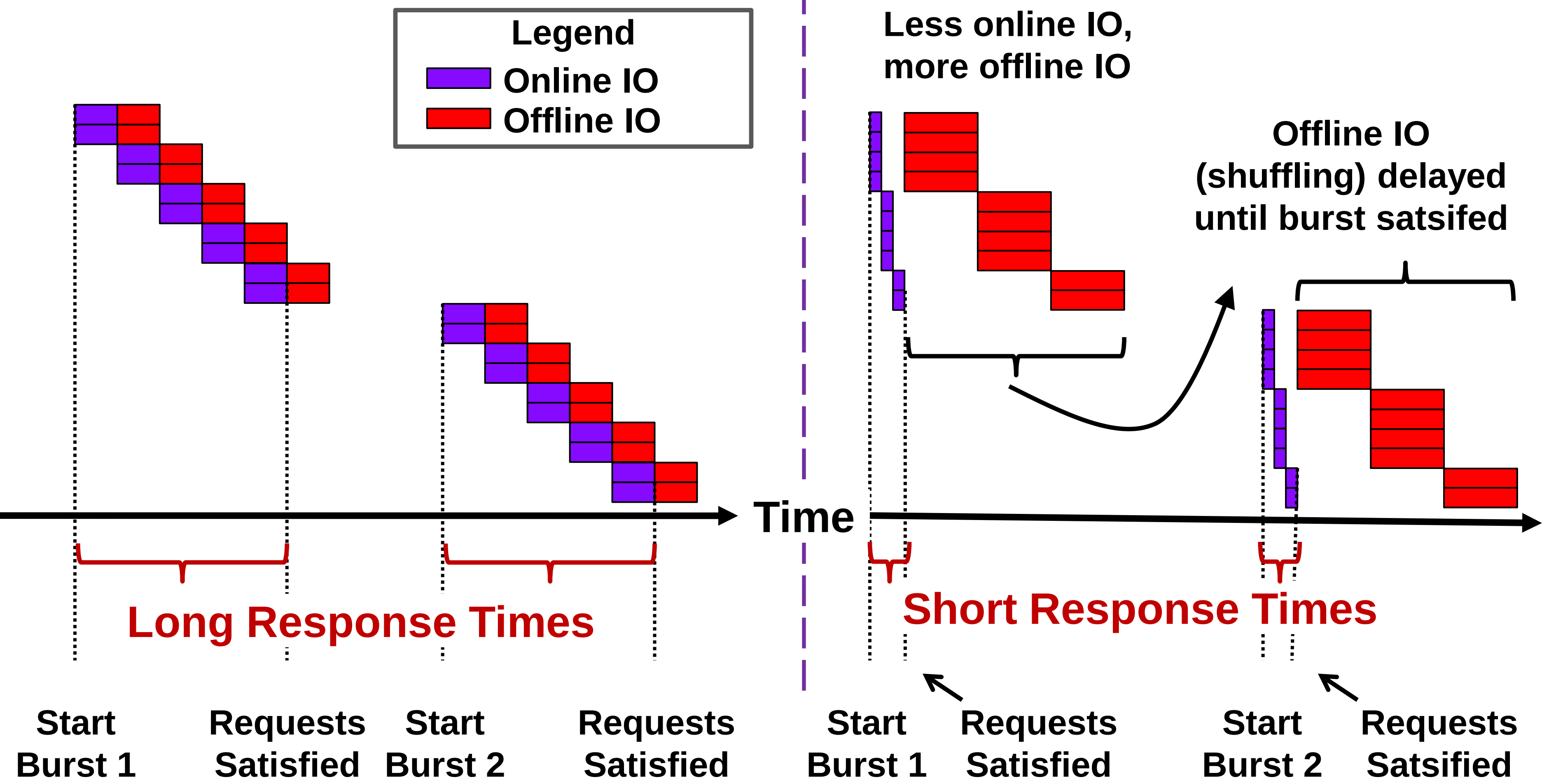
- Maximally utilize client space

## 3: Prioritize Efficient Shuffling

- Less work to free same client space
- Efficiency ≈ Space Freed / Required IO
- Scheduling policy must be “oblivious”

# Existing ORAMs

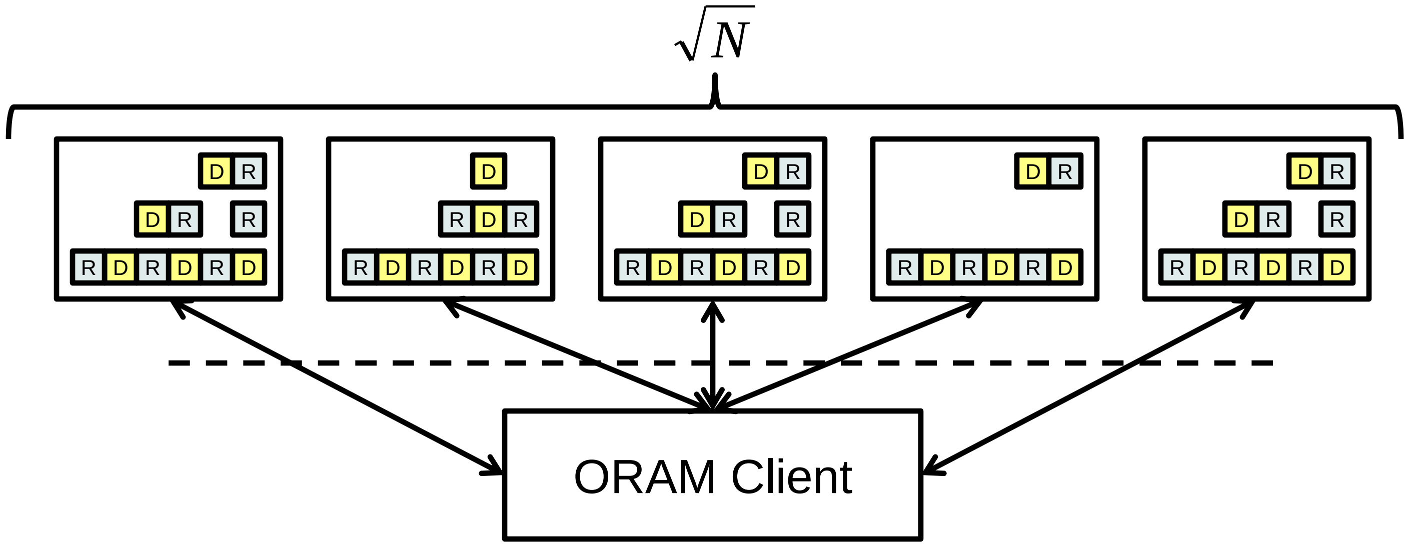
# Burst ORAM



# Outline

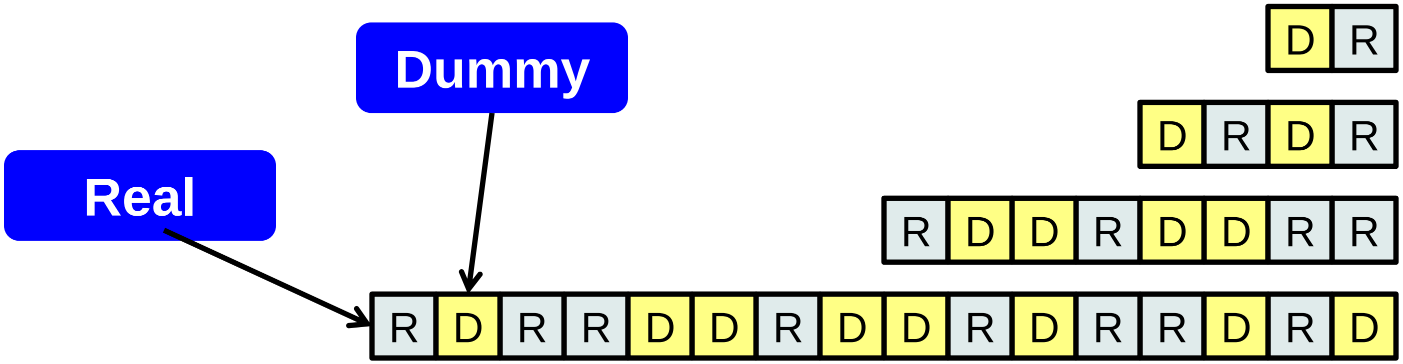
- Private Data Outsourcing
  - Oblivious RAM
- Practical ORAM Response Times
  - Burst ORAM
- Burst ORAM Details
- Results

# Layout: Burst ORAM & ObliviStore




- $\sqrt{N}$ partitions of $\sqrt{N}$ blocks each
- Each request routed to single partition

# ObliviStore ORAM (Stefanov et al. 2013)




Online Cost
(Blocks Transferred During Read)

Offline Cost
(Blocks Transferred During Shuffling)

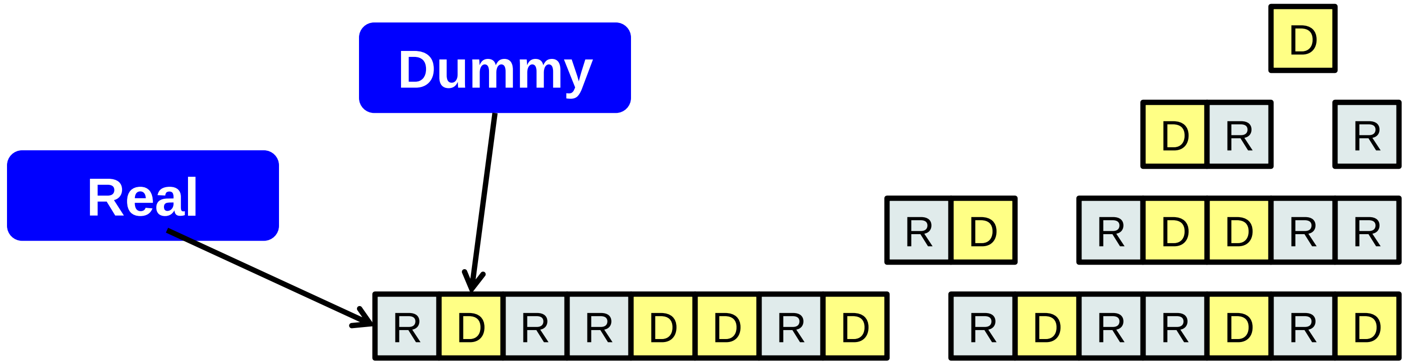
Req. 1

Req. 2

Req. 3

Idle Time

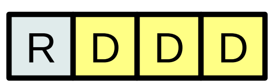
# ObliviStore ORAM (Stefanov et al. 2013)




Online Cost
(Blocks Transferred During Read)

Offline Cost
(Blocks Transferred During Shuffling)

Req. 1

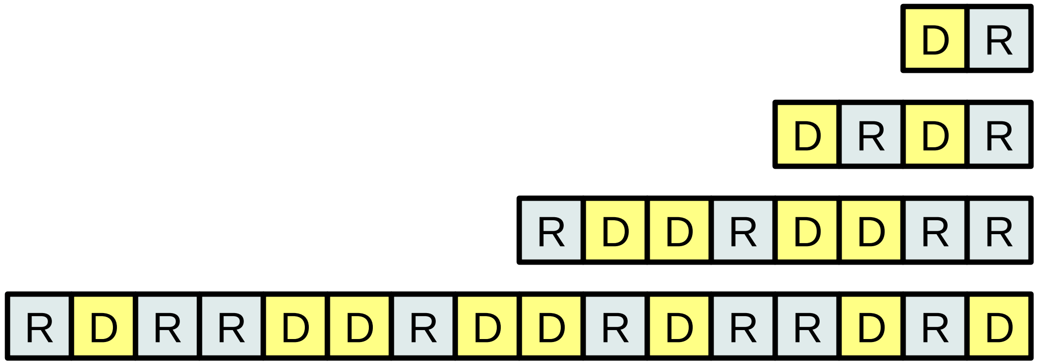



Req. 2

Req. 3

Idle Time

# ObliviStore ORAM (Stefanov et al. 2013)




Online Cost
(Blocks Transferred During Read)

Offline Cost
(Blocks Transferred During Shuffling)

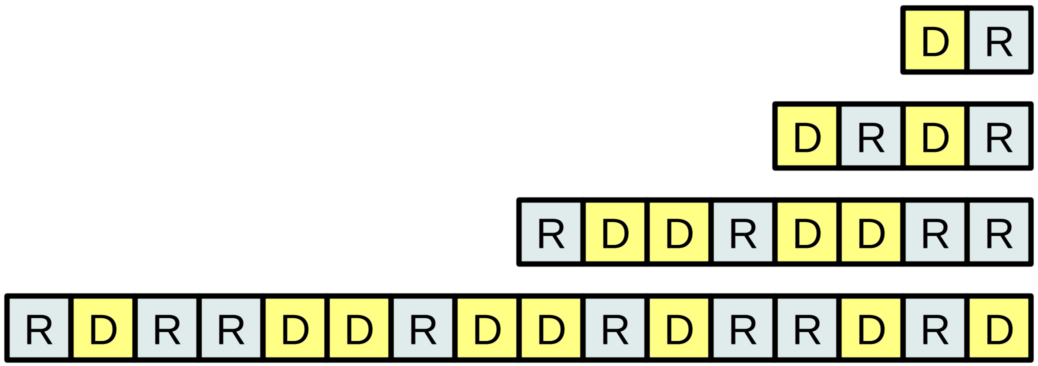
Req. 1


Req. 2

Req. 3

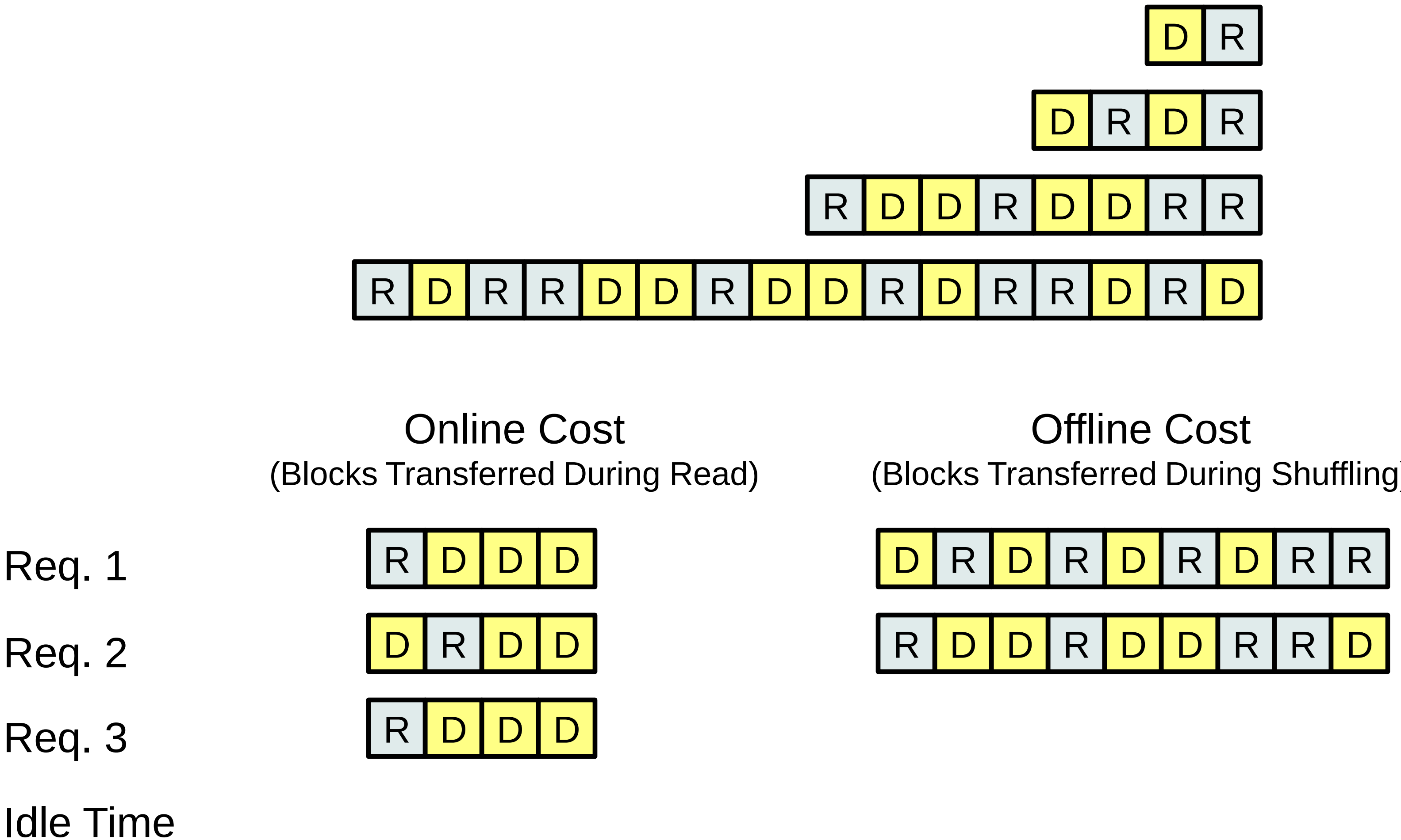
Idle Time

# ObliviStore ORAM (Stefanov et al. 2013)

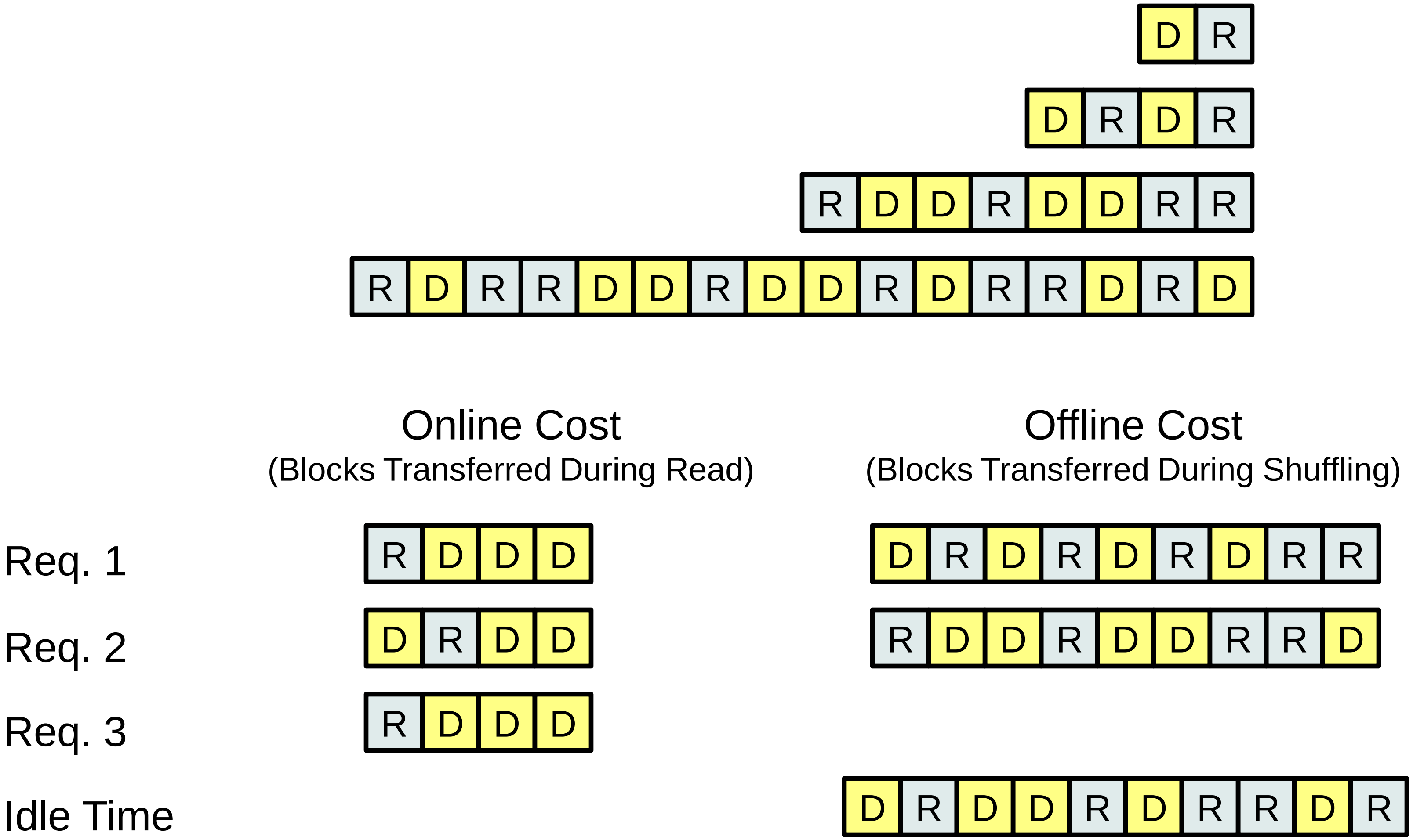



| | Online Cost (Blocks Transferred During Read) | Offline Cost (Blocks Transferred During Shuffling) |
|---|---|---|
| Req. 1 | R D D D | D R D R D R D R R |
| Req. 2 | D R D D | R D D R D D R R D |
| Req. 3 | | |
| Idle Time | | |

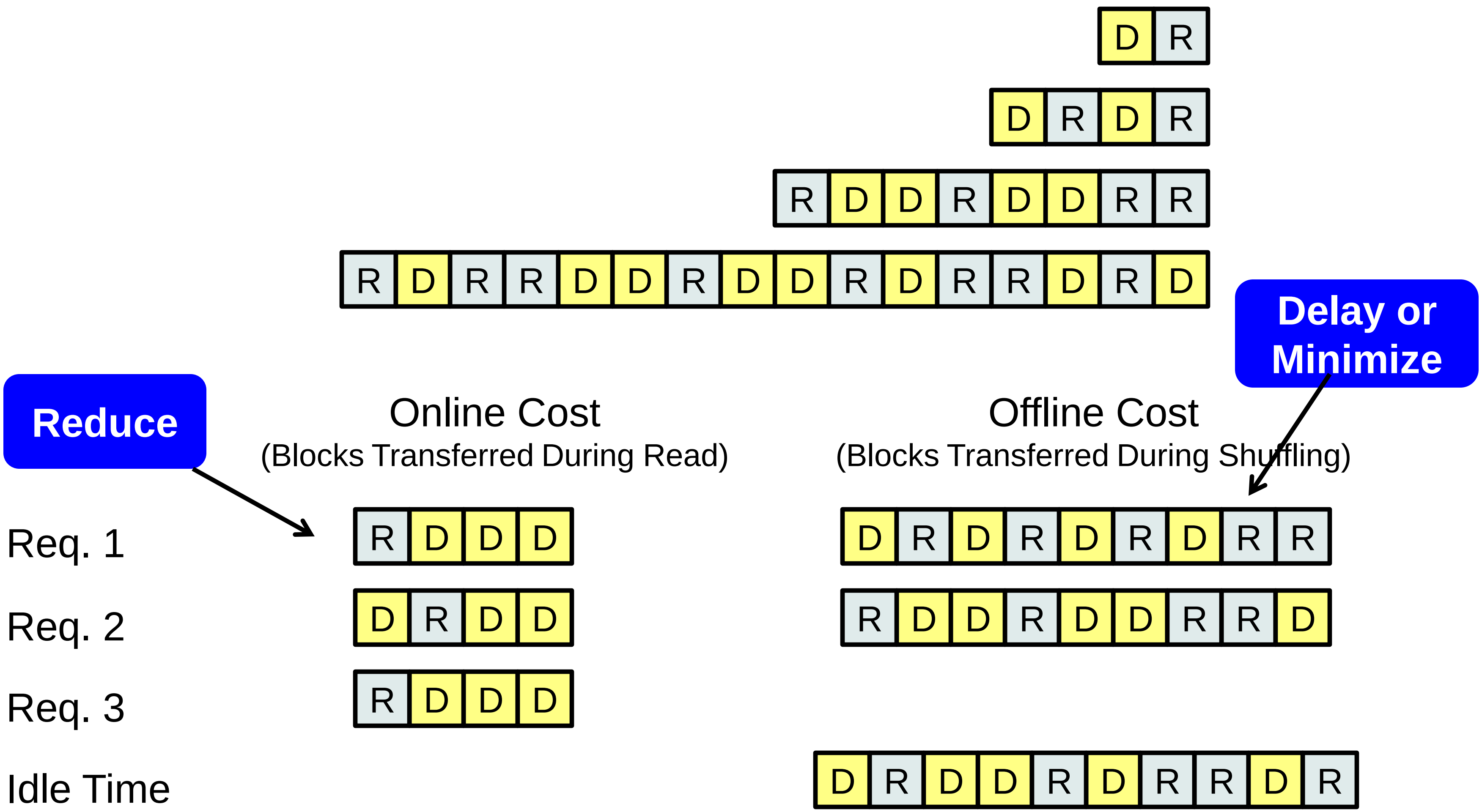
# ObliviStore ORAM (Stefanov et al. 2013)



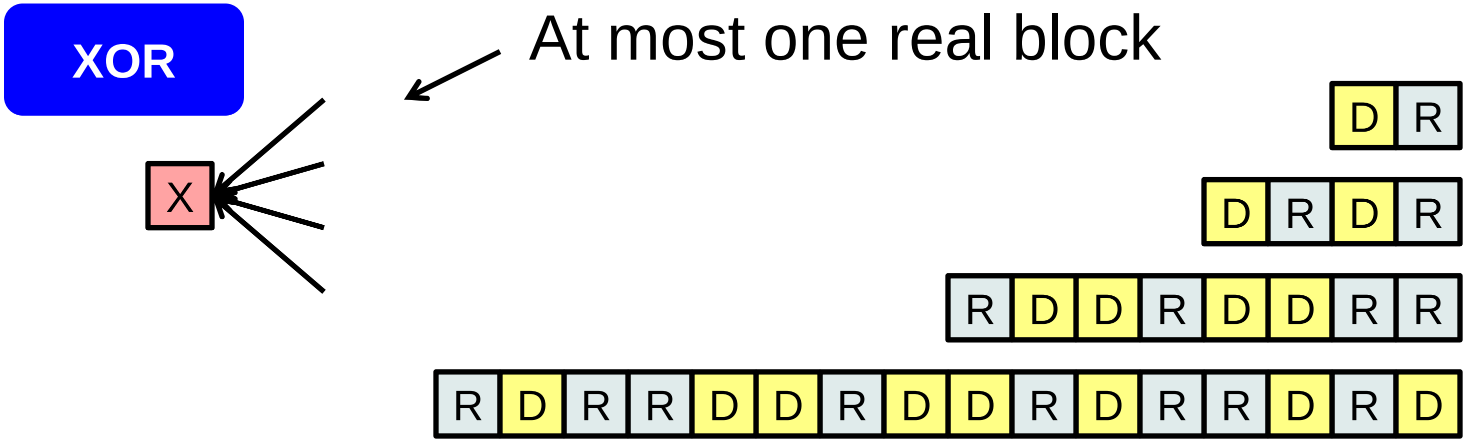
# ObliviStore ORAM (Stefanov et al. 2013)



# Burst ORAM


D R
D R D R
R D D R D D R R
R D R R D D R D D R D R R D R D
Delay or Minimize
Reduce
Online Cost
(Blocks Transferred During Read)
Offline Cost
(Blocks Transferred During Shuffling)
Req. 1
R D D D
D R D R D R D R R
Req. 2
D R D D
R D D R D D R R D
Req. 3
R D D D
Idle Time
D R D D R D R R D R

# Burst ORAM: XOR Technique




Online Cost
(Blocks Transferred During Read)

Offline Cost
(Blocks Transferred During Shuffling)

Req. 1

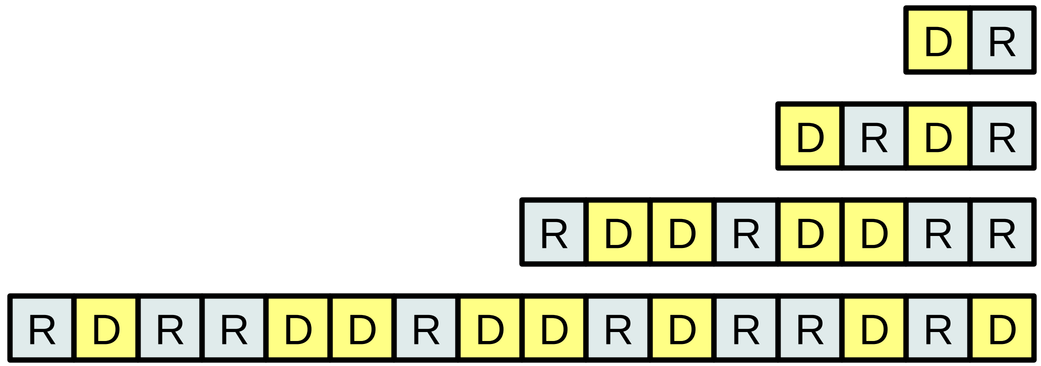
Req. 2

Req. 3

Idle Time

Dummy blocks reconstructed locally and subtracted out

# Burst ORAM




Online Cost
(Blocks Transferred During Read)

Offline Cost
(Blocks Transferred During Shuffling)

Req. 1 X

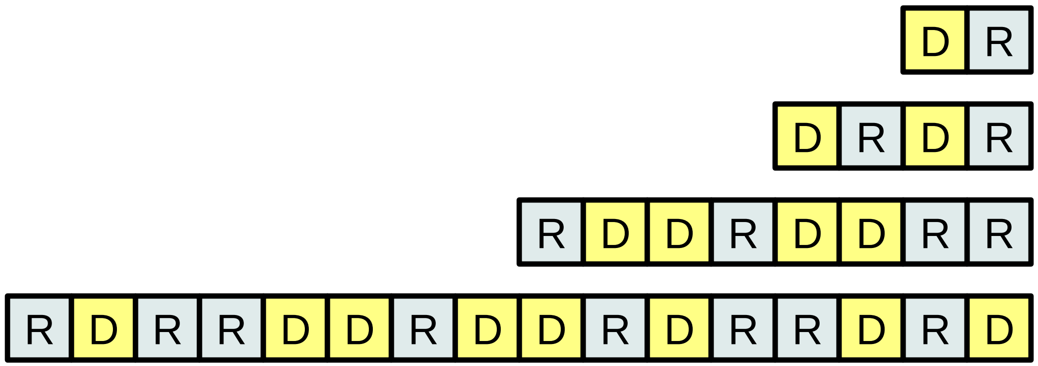
Req. 2 X D R

Req. 3

Idle Time

**Small Amount of Shuffling to Free Space for Next Block on Client**

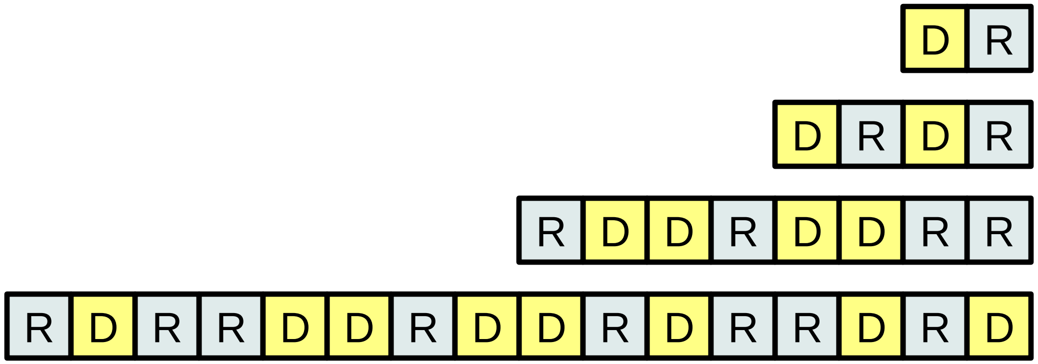
# Burst ORAM



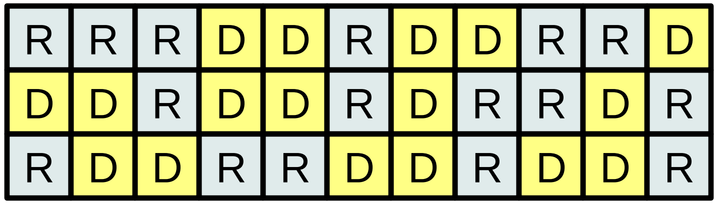

| | Online Cost (Blocks Transferred During Read) | Offline Cost (Blocks Transferred During Shuffling) |
|---|---|---|
| Req. 1 | X | |
| Req. 2 | X | D R |
| Req. 3 | X | |
| Idle Time | | |

# Burst ORAM

**Well suited to mobile device**



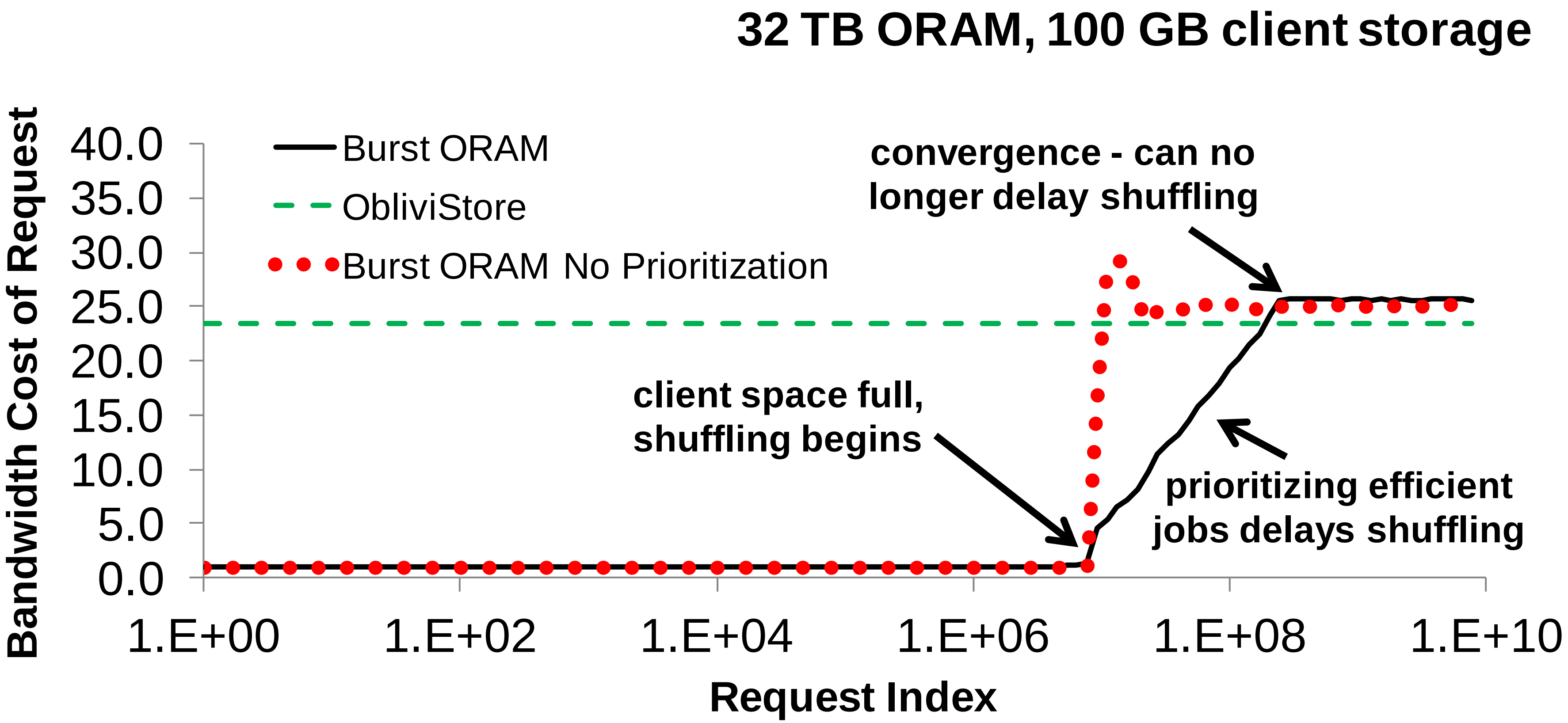

| | Online Cost (Blocks Transferred During Read) | Offline Cost (Blocks Transferred During Shuffling) |
|---|---|---|
| Req. 1 | X | |
| Req. 2 | X | D R |
| Req. 3 | X | |
| Idle Time | | |



# Outline

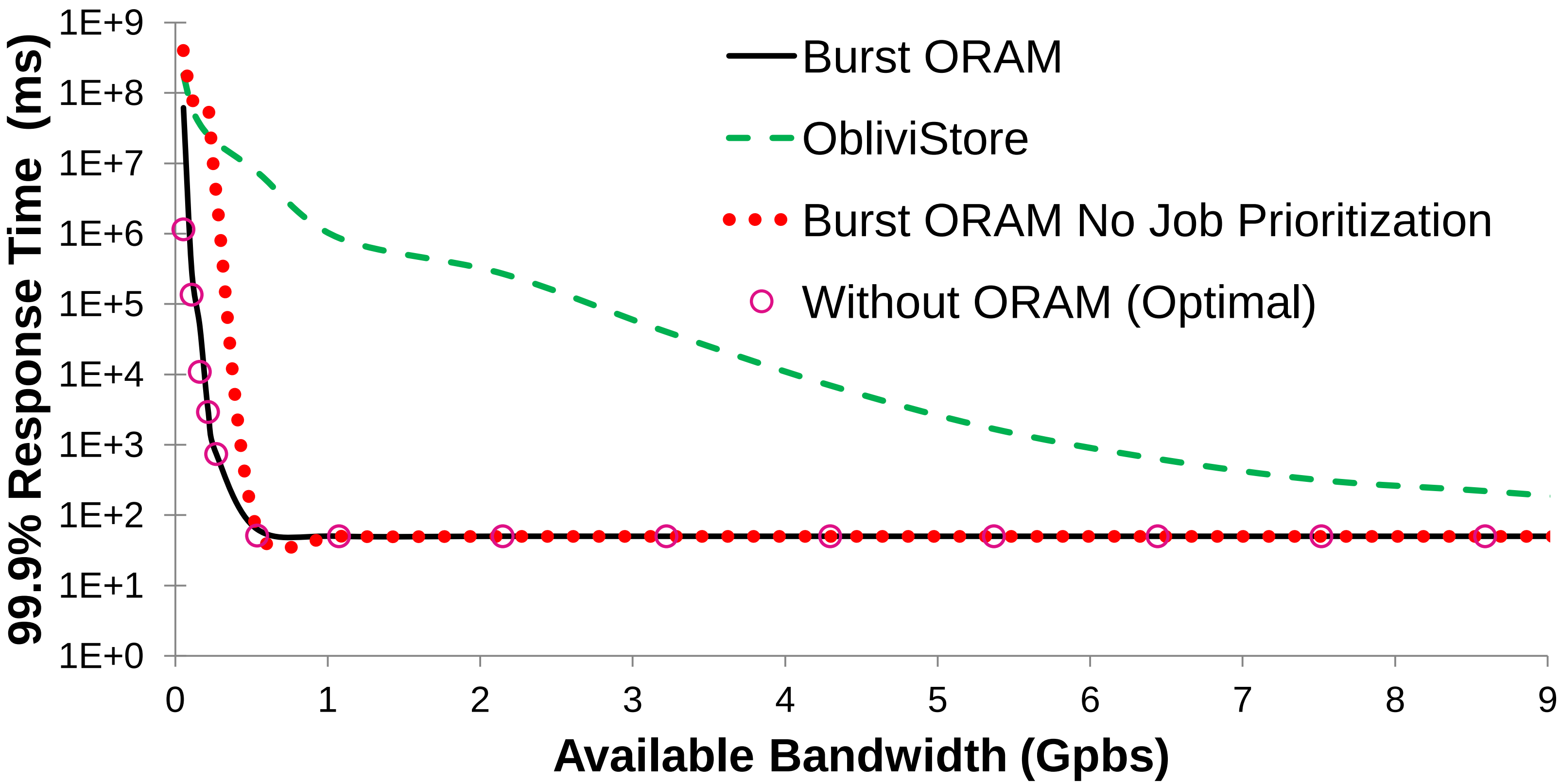
- Private Data Outsourcing
  - Oblivious RAM
- Practical ORAM Response Times
  - Burst ORAM
- Burst ORAM Details
- Results

# Burst ORAM – Extended Burst



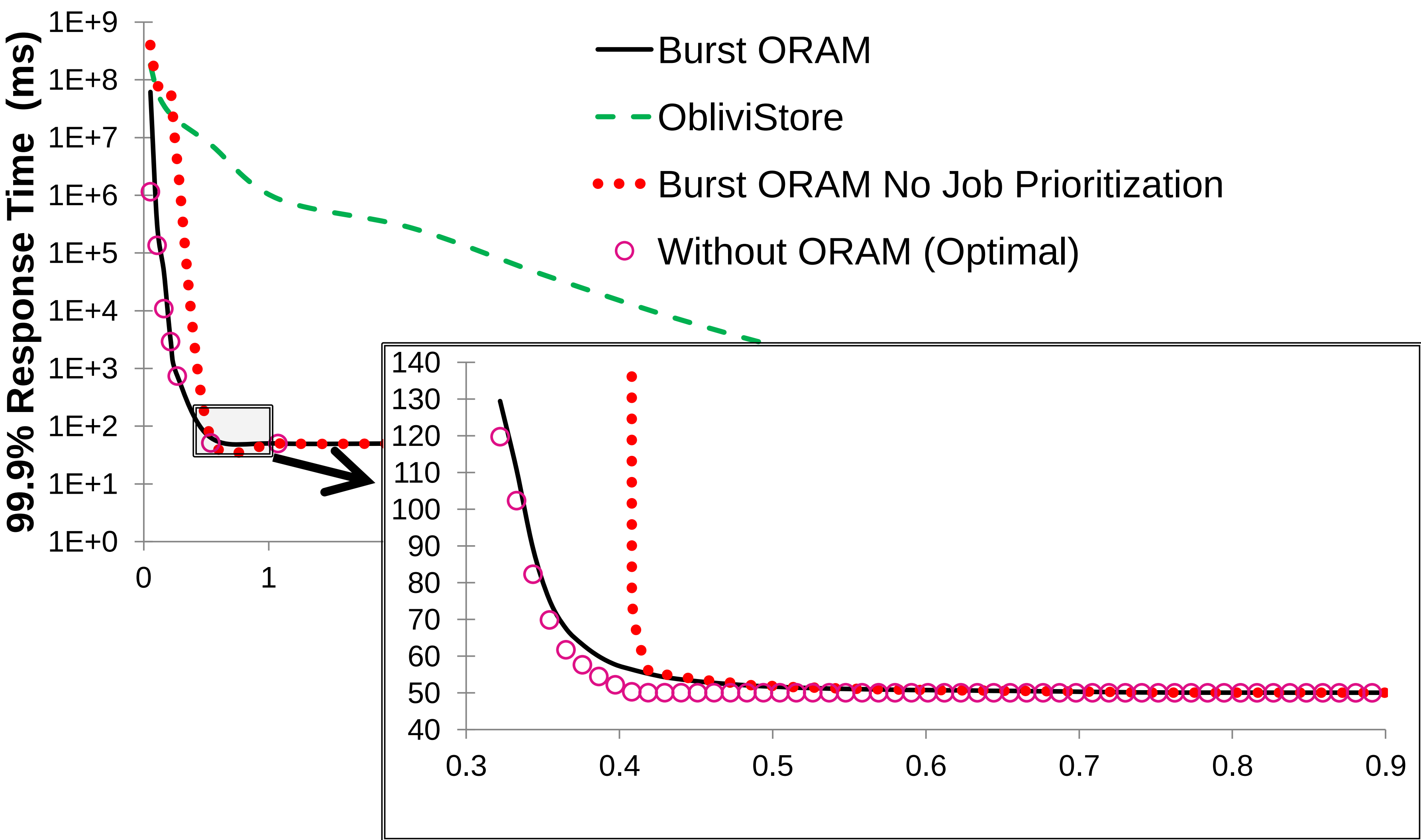
# 99.9% Reponse Time Comparison on NetApp Trace

(50ms network latency, 32 TB ORAM, 100 GB client storage)



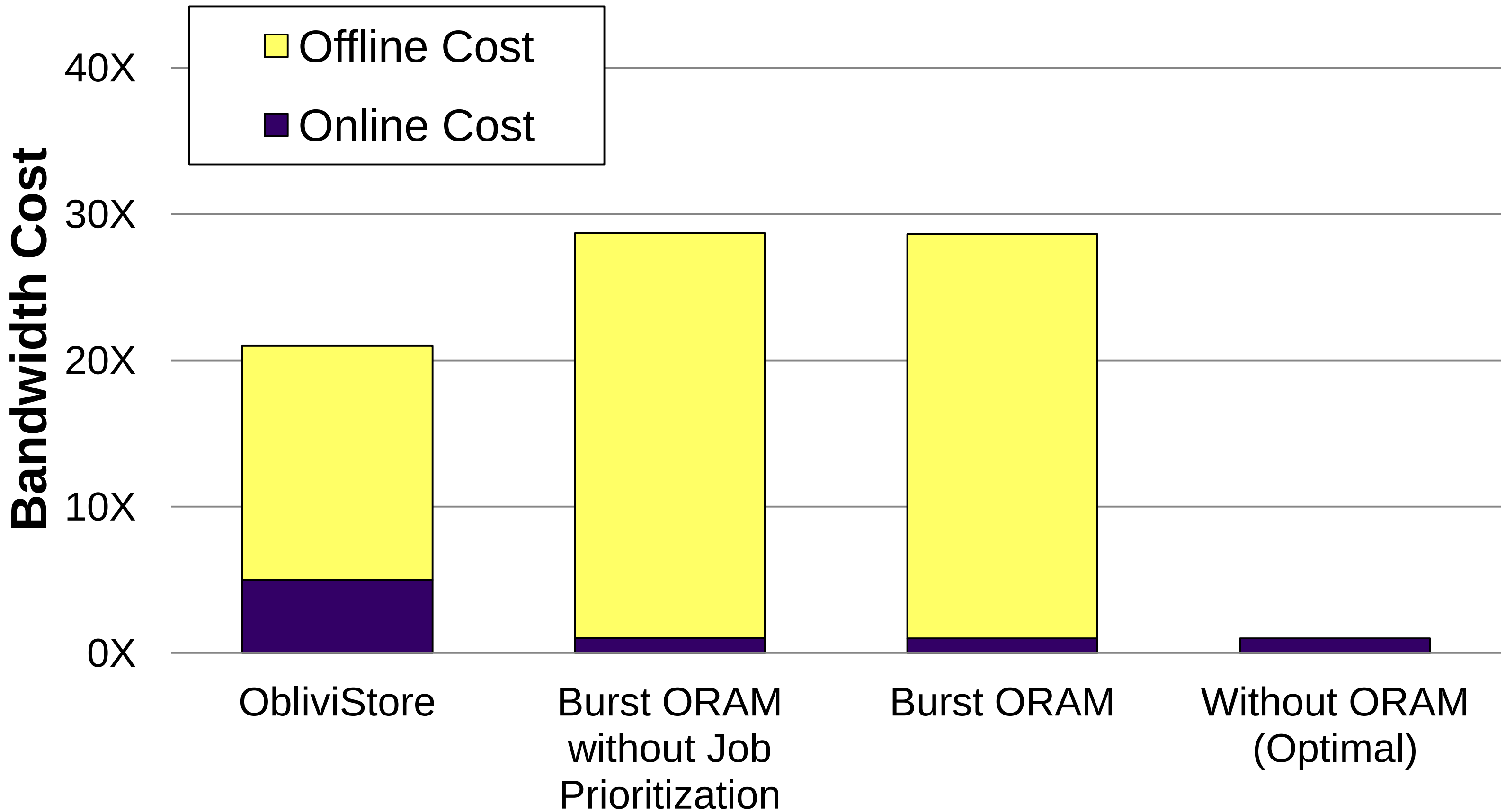
# 99.9% Reponse Time Comparison on NetApp Trace

(50ms network latency, 32 TB ORAM, 100 GB client storage)



**NetApp Trace Bandwidth Costs**

(50ms network latency, 32 TB ORAM, 100 GB client storage, 400Mbps bandwidth)



# Conclusion

- Accomplishments
  - Practical response times during bursts
  - Maintains low bandwidth cost
- Limitations
  - Does not reduce *total* bandwidth cost
- Future
  - Lower bandwidth cost and low response times

# ?