8th International Conference on
Autonomous Infrastructure, Management and Security

# ESPRESSO: An Encryption as a Service for Cloud Storage Systems

Kang Seungmin

30th Jun., 2014

# Outline

NUS National University of Singapore

- **Introduction and motivation**
- **Main contribution**
- **Detailed proposed solution**
  - Cloud storage system models
  - Overall architecture of ESPRESSO
  - Implementation of ESPRESSO
  - Integration of ESPRESSO into Swift and Cumulus
- **Experiments and performance evaluation**
- **Conclusion and future work.**

# Introduction and Motivation

- **Cloud storage systems provide high data availability and the flexibility in data management, and they become the primary storage space for cloud users' data.**
- **Data privacy is one of the most important challenges to be solved due to the shared storage space characteristic of cloud storage systems.**
- **Data encryption emerged as one of the most effective means to protect sensitive data.**
- **Among existing CSPs, only Google Cloud Storage and Amazon S3 provide such encryption service.**
- **Many other CSPs do not have yet this service.**

# Main Contribution

- **Proposing a solid encryption service which allows any CSP to integrate it**
  - Designing and implementing an encryption service, called ESPRESSO (Encryption as a Service for Cloud Storage Systems).

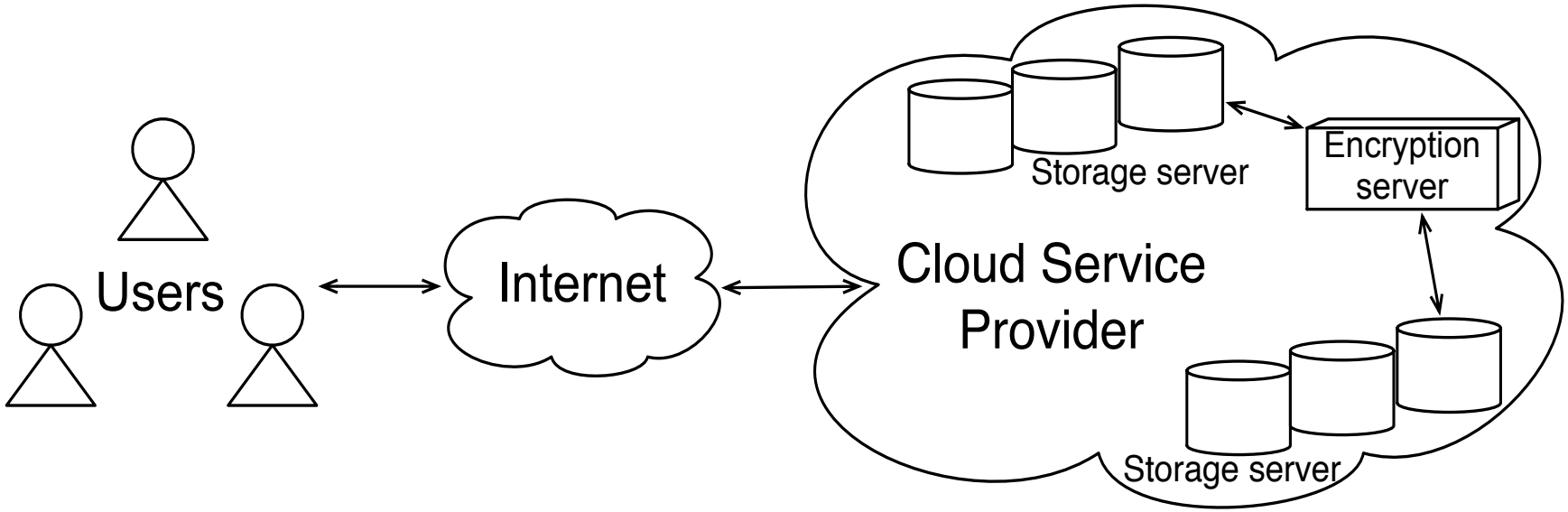
- **ESPRESSO**
  - is a standalone service
  - is configurable and flexible service for both CSPs and cloud users
    - **CSPs can choose the encryption algorithm based on their preference**
    - **Users can specify the critical level of their data**
  - is easily integrated.

# Outline

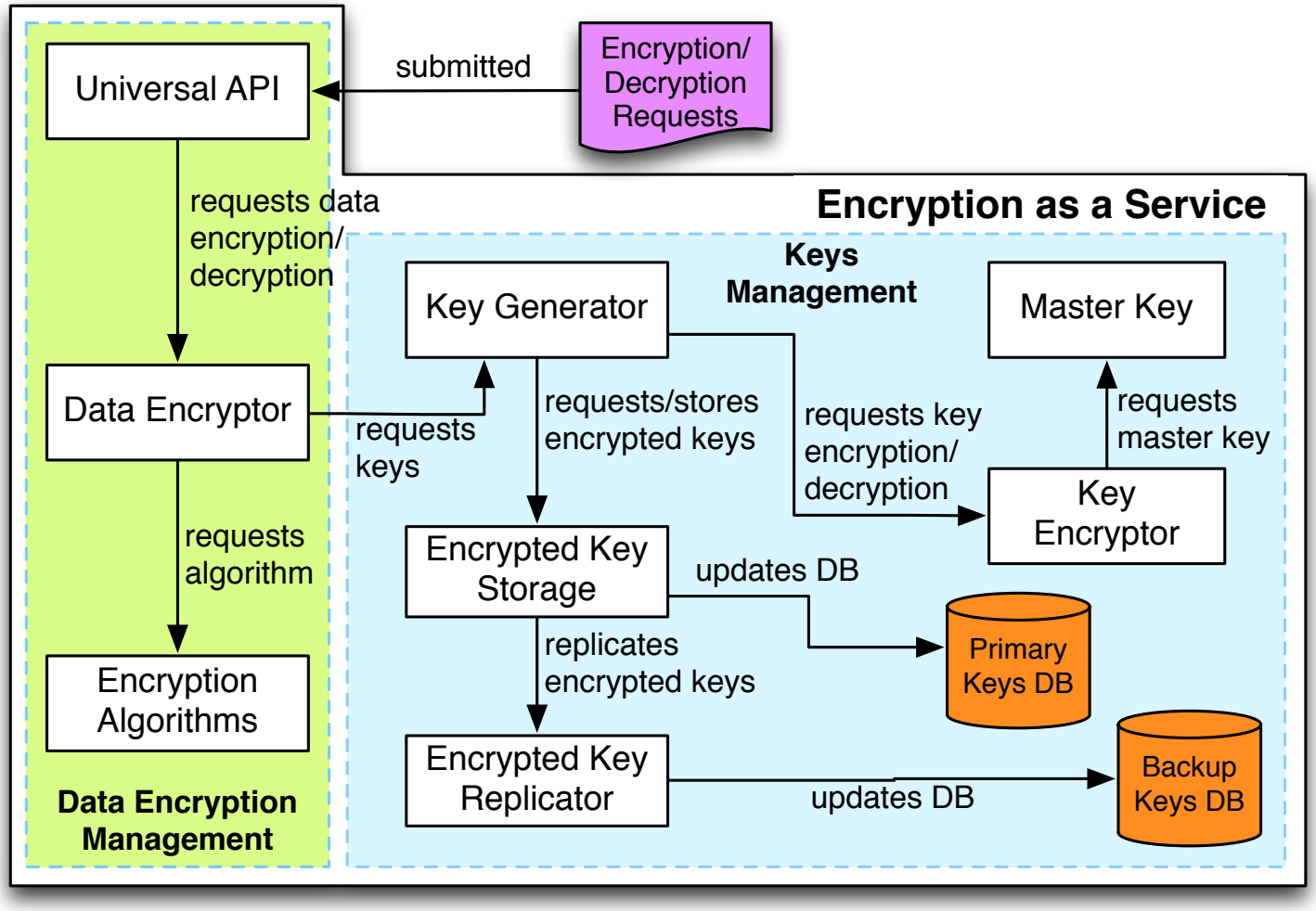
- Introduction and motivation
- Main contribution
- **Detailed proposed solution**
  - Cloud storage system models
  - Overall architecture of ESPRESSO
  - Implementation of ESPRESSO
  - Integration of ESPRESSO into Swift and Cumulus
- Experiments and performance evaluation
- Conclusion and future work.

# Cloud Storage System Models

- **Cloud users deploy the encryption software on their local machine or on a remote machine in their trusted domain.**
- **Cloud users rely on a third party who deploys the encryption software and provides it to users as a service.**
- **CSPs deploy the encryption software on a server in its trusted domain as one of its components.**

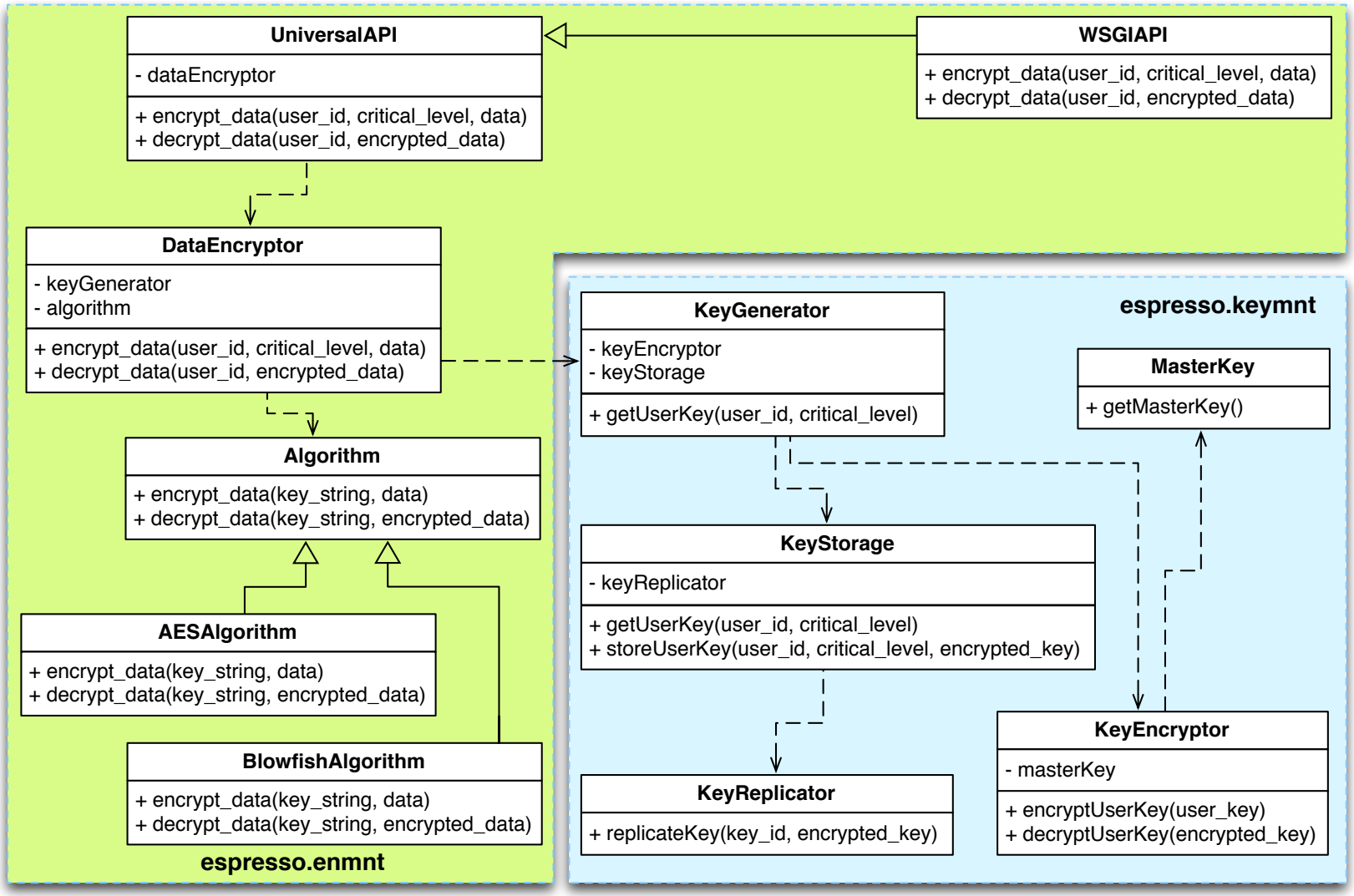


# Overall Architecture of ESPRESSO



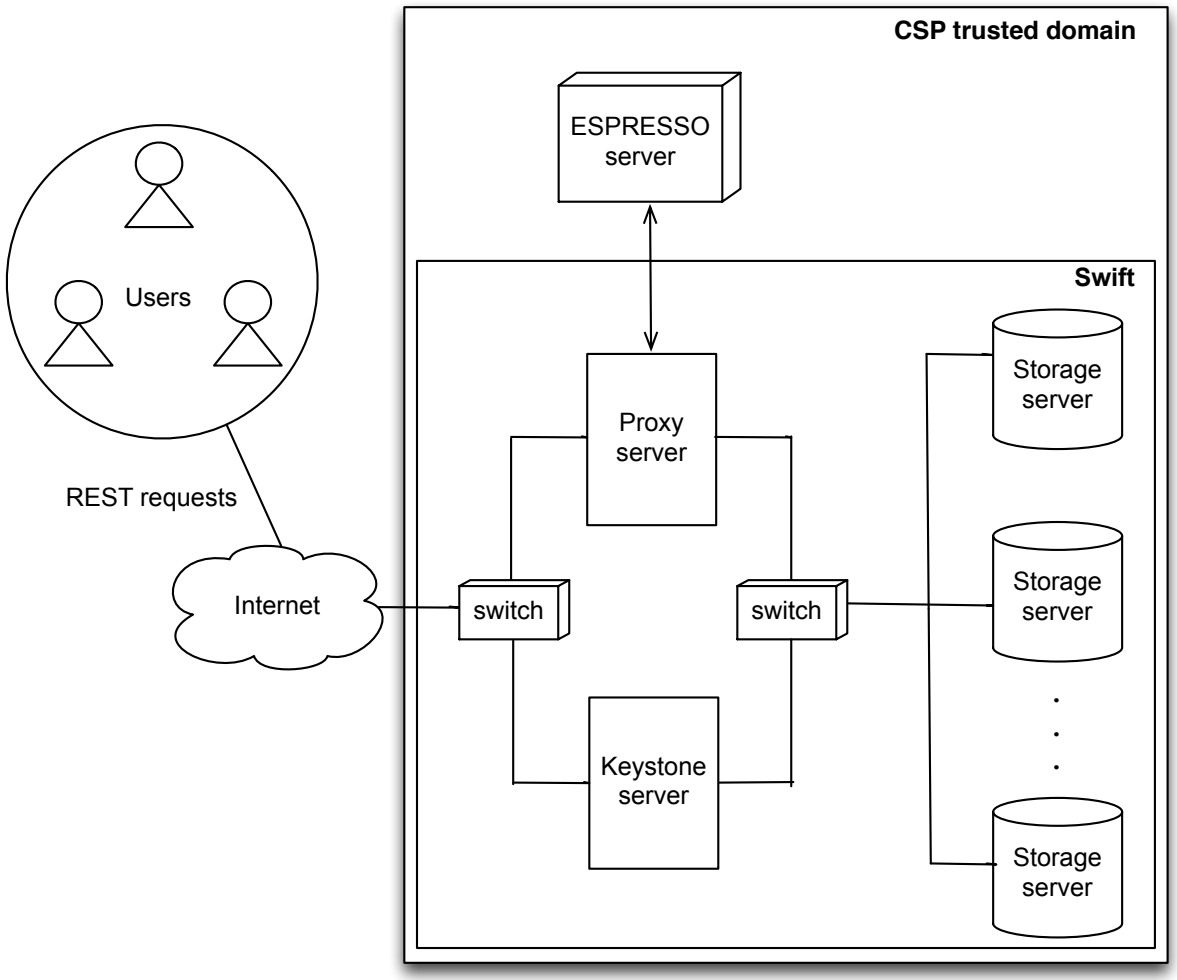

- **Two main components**
  - Data encryption management.
  - Key management.
- **Supporting flexibility in ESPRESSO**
  - AES and Blowfish algorithms.
  - Three critical levels with three key lengths: 128, 192 and 256 bits.

# Implementation of ESPRESSO



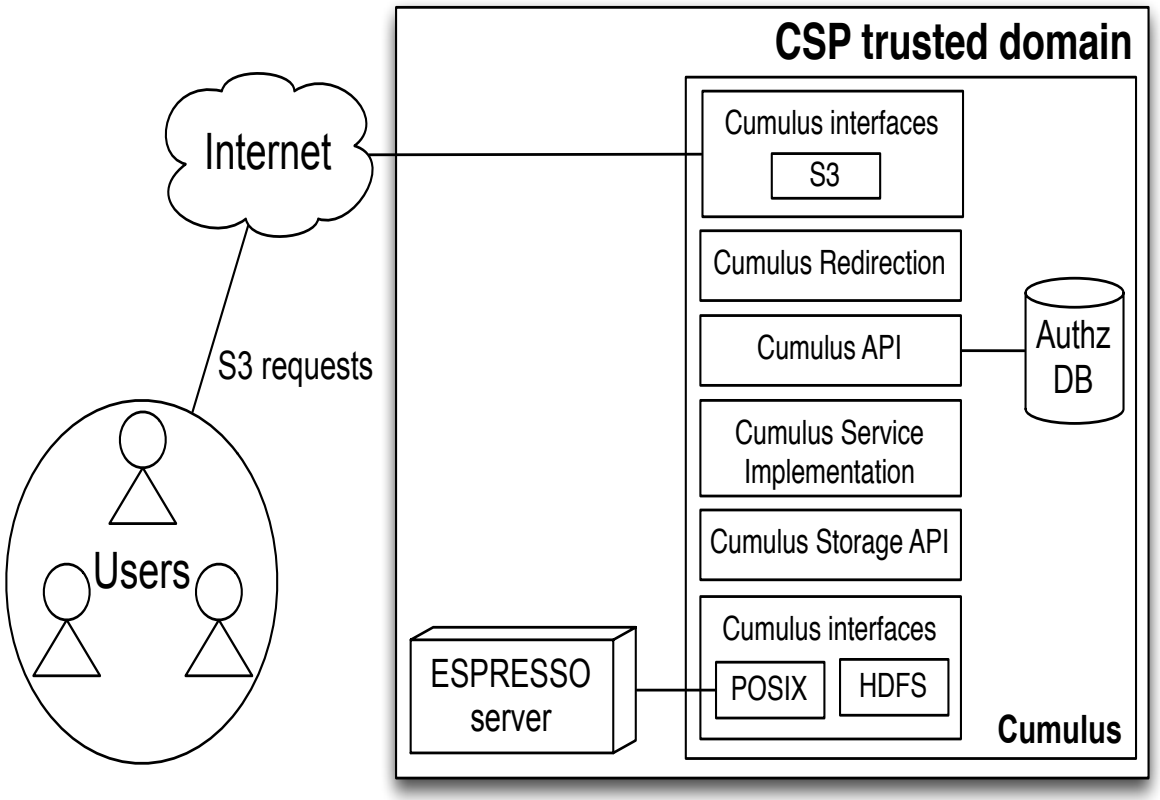

- **Implementing the abstract Universal API class to allow multiple CSPs to integrate ESPRESSO, e.g., SwiftAPI.**
- **Implementing the abstract Algorithm class to support different encryption algorithms.**

# Integration of ESPRESSO into Swift




- **ESPRESSO is deployed on a separate server.**
- **The proxy server initializes encryption on ESPRESSO.**
- **All modification were made in swift/proxy/controller/obj.py.**
- **Less than 50 code lines were added for encryption and decryption requests in Swift.**
- **Parameters of requests:**
  - Input data
  - User identification
  - Critical level of data

# Integration of ESPRESSO into Cumulus




- **ESPRESSO is deployed on a separate server.**
- **The Cumulus interface initializes encryption on ESPRESSO.**
- **All modification were made in cumulus/cb/pycb/cbRequest.py.**
- **Less than 50 code lines were added for encryption and decryption requests in Cumulus.**
- **--add-header “critical level: A”.**

# Outline

- **Introduction and motivation**
- **Main contribution**
- **Detailed proposed solution**
  - Cloud storage system models
  - Overall architecture of ESPRESSO
  - Implementation of ESPRESSO
  - Integration of ESPRESSO into Swift and Cumulus
- **Experiments and performance evaluation**
- **Conclusion and future work.**

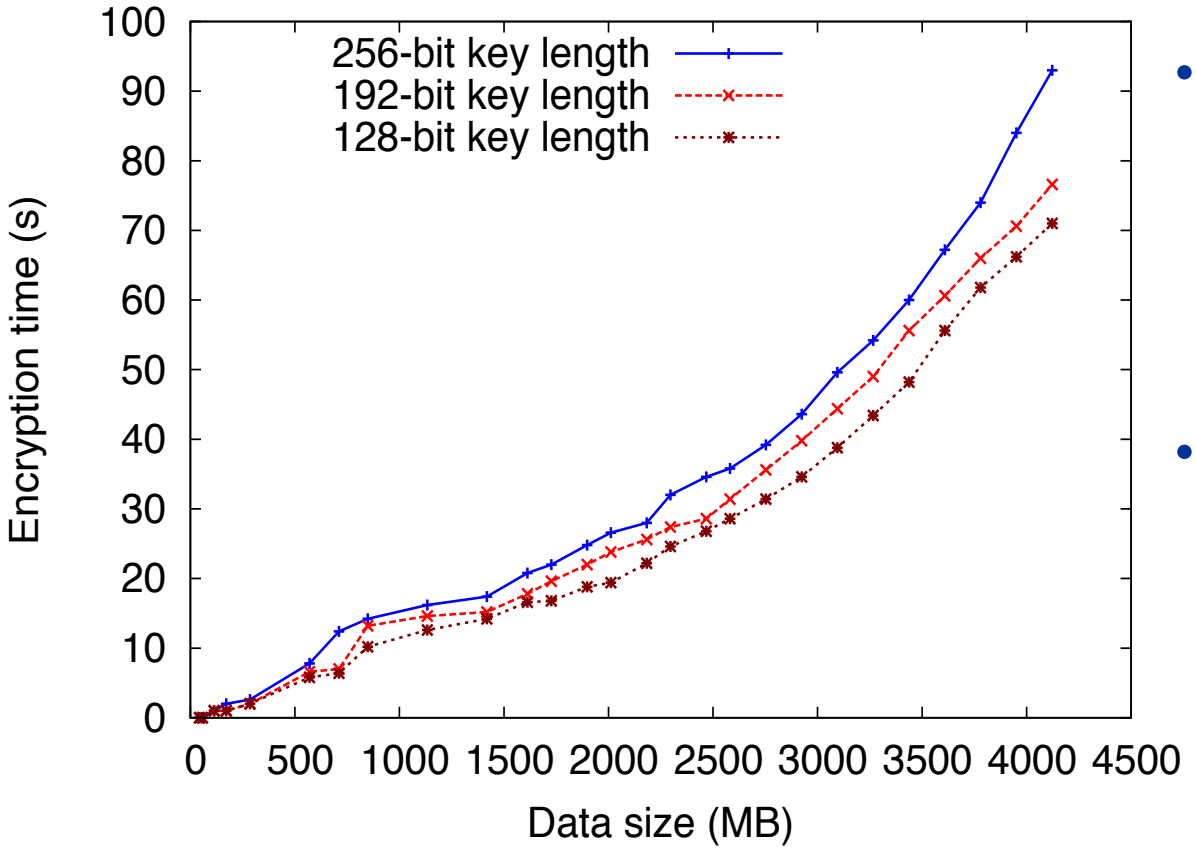
# Experiments and Performance Evaluation

- **Experiment setup**
  - Deploying the integrated Swift/Cumulus storage system on two dedicated physical servers of the same rack
    - **PowerEdge C6220 with Intel(R) Xeon(R) Processor E5-2640 2.50GHz, 24GB RAM**
  - Using real data files which are downloaded from the Wikipedia archive. Data size varies from 100MB to 4000MB (~4GB).

- **Performance metrics**
  - Latency of encryption algorithms
  - Latency of the storage system with and without ESPRESSO
  - Impact of network bandwidth
  - Comparison of Swift and Cumulus.

# Experiments and Performance Evaluation

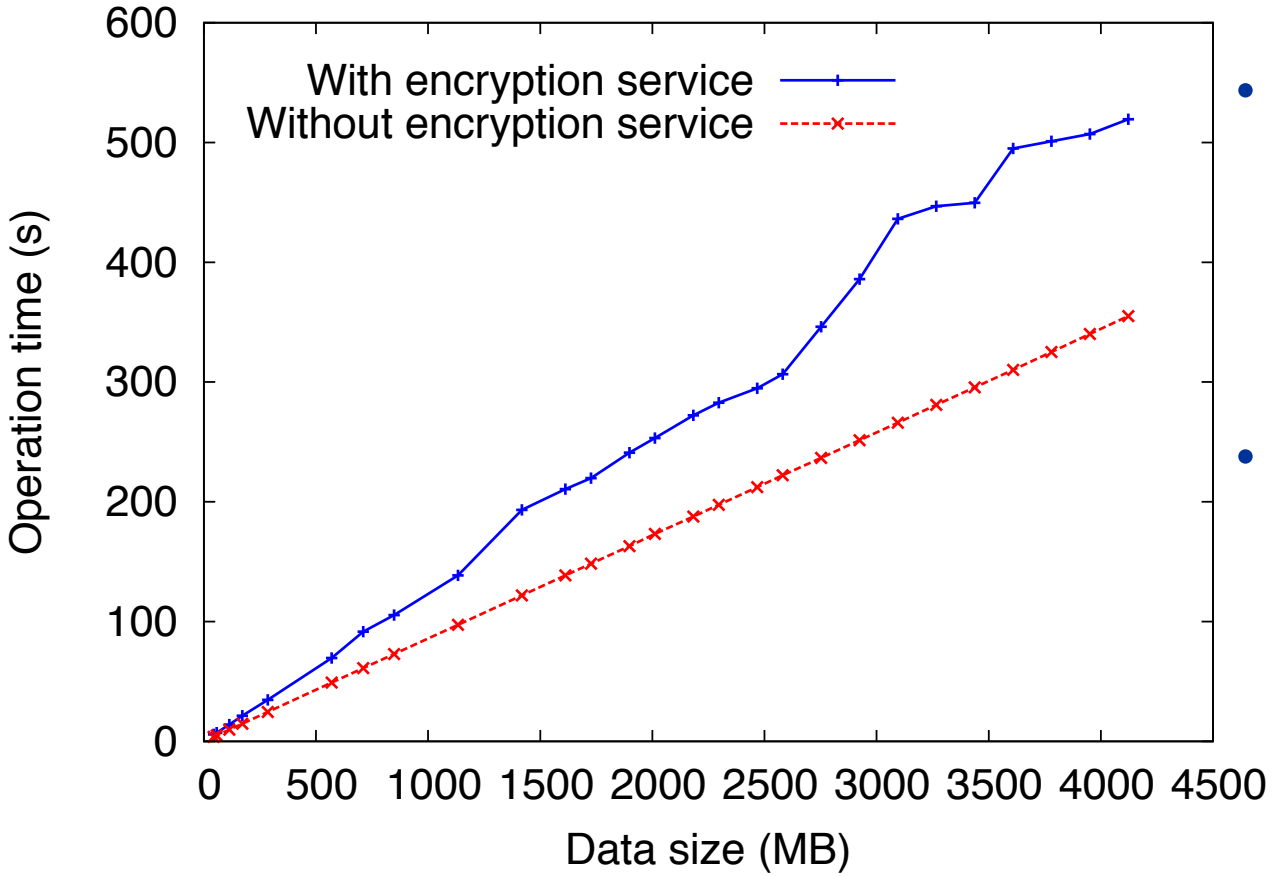
- **Latency of encryption algorithm**




- **With the same key length, the larger data volume, the longer time needed to complete the encryption.**
- **The longer key provides higher security level, however, needs longer time to complete the encryption.**

# Experiments and Performance Evaluation

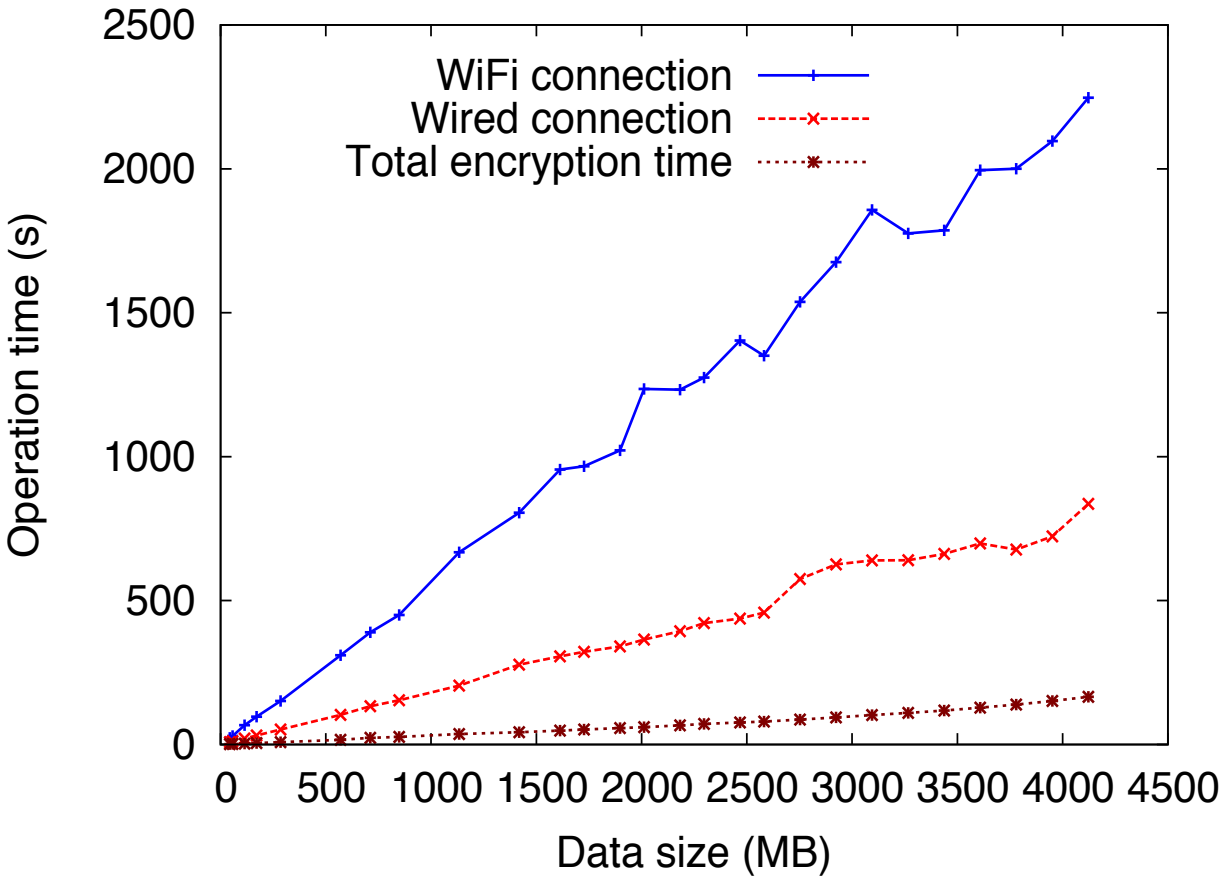
- **Overall latency with and without ESPRESSO**




- **Without ESPRESSO**
  - The total time is considered the data transfer time from the client to the Swift server.
- **With ESPRESSO**
  - Additional overhead includes the encryption time and the data transfer time between the Swift and ESPRESSO servers.

# Experiments and Performance Evaluation

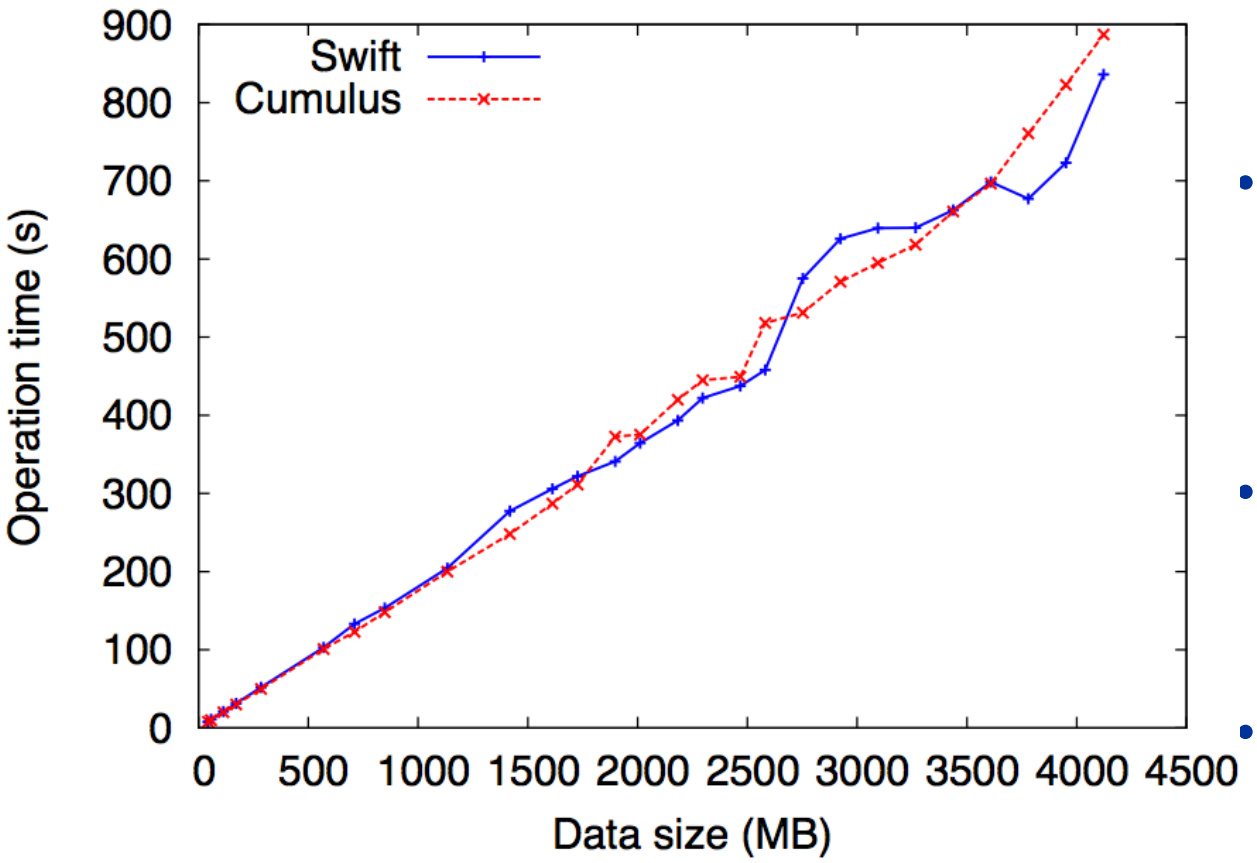
- **Impact of network bandwidth**




- **The client machine is 3kms far from the Swift server.**
- **The data transfer time dominates in both cases:**
  - WiFi (~2 Mbps)
  - Wired connection (~10 Mbps).
- **The observed encryption time overhead is negligible (2.75 mins) compared to the total uploading time (37.45 mins) with the WiFi connection.**

# Experiments and Performance Evaluation

- **Swift vs. Cumulus performance**




- **The operation times of both systems are almost the same.**
- **Swift needs longer time for replicating data with three copies.**
- **Cumulus does not provide the replication service.**
- **The overhead on Swift is compromised by the fluctuation of data transfer time.**

# Conclusion and Future Work

- **We provided ESPRESSO, an encryption service which is**
  - Standalone
  - Transparent
  - Flexible
- **Real experiments assess the performance and effectiveness of ESPRESSO.**
- **Any CSP can integrate ESPRESSO into its infrastructure without heavy modification.**
- **Future Work: integrate ESPRESSO with Homomorphic encryption (HE).**

# Thank you for your attention!

# Q & A