Delegated Access for Hadoop Clusters in the Cloud

David Nuñez, Isaac Agudo, and Javier Lopez

Network, Information and Computer Security Laboratory (NICS Lab)
Universidad de Málaga, Spain
Email: dnunez@lcc.uma.es

IEEE CloudCom 2014 – Singapore

1. Introduction
   - Motivation
   - Scenario
   - Proposal
2. The Hadoop Framework
3. Proxy Re-Encryption
   - Overview
   - Access Control based on Proxy Re-Encryption
4. DASHR: Delegated Access System for Hadoop based on Re-Encryption
5. Experimental results
   - Experimental setting
   - Results
6. Conclusions

Introduction

- **Big Data** $\Rightarrow$ use of vast amounts of data that makes processing and maintenance virtually impossible from the traditional perspective of information management
- **Security** and **Privacy** challenges
  - In some cases the data stored is sensitive or personal
  - Malicious agents (insiders and outsiders) can make a profit by selling or exploiting this data
  - Security is usually delegated to access control enforcement layers, which are implemented on top of the actual data stores.
  - Technical staff (e.g., system administrators) are often able to bypass these traditional access control systems and read data at will.

Apache Hadoop

- **Apache Hadoop** stands out as the most prominent framework for processing big datasets.
- Apache Hadoop is a framework that enables the storing and processing of large-scale datasets by clusters of machines.
- The strategy of Hadoop is to divide the workload into parts and spreading them throughout the cluster.
- Hadoop was not designed with security in mind
- However, it is widely used by organizations that have strong security requirements regarding data protection.

Goal

- **Goal** $\Rightarrow$ Stronger safeguards based on **cryptography**
- Cloud Security Alliance on Security Challenges for Big Data : "*[...] sensitive data must be protected through the use of cryptography and granular access control*".
- **Motivating Scenario** $\Rightarrow$ Big Data Analytics as a Service

Motivating Scenario: Big Data Analytics as a Service

- Big Data Analytics is a new opportunity for organizations to transform the way they market services and products through the analysis of massive amounts of data
- Small and medium size companies are not often capable of acquiring and maintaining the necessary infrastructure for running Big Data Analytics on-premise
- The Cloud is a natural solution to this problem, in particular for small organizations
- Access to on-demand high-end clusters for analysing massive amounts of data (e.g.: Hadoop on Google Cloud)
- It has even more sense when the organizations are already operating in the cloud, so analytics can be performed where the data is located

Motivating Scenario: Big Data Analytics as a Service

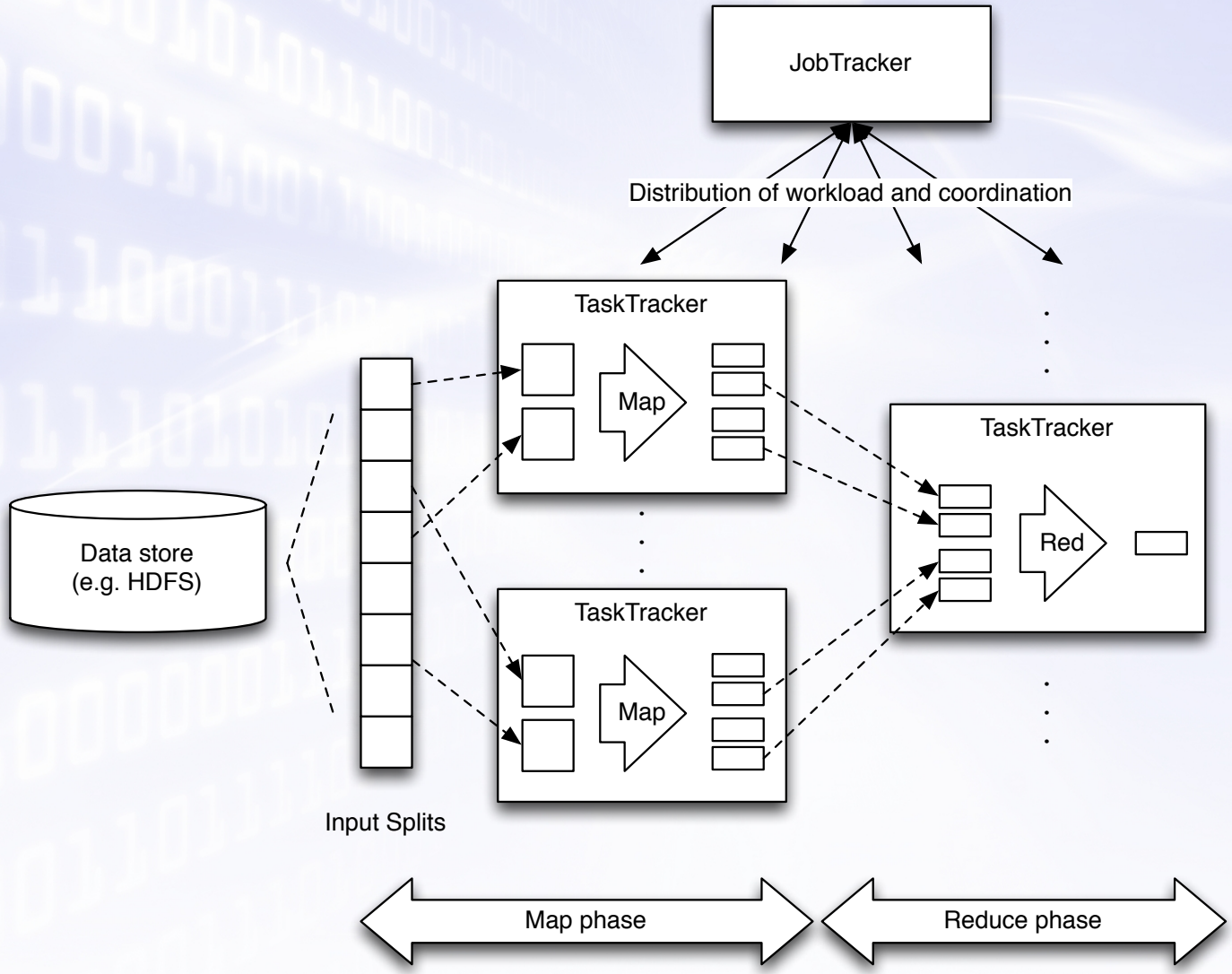
- There are several risks, such as the ones that stem for a multi-tenant environment. Jobs and data from different tenants are then kept together under the same cluster in the cloud, which could be unsafe when one considers the weak security measures provided by Hadoop
- The use of encryption for protecting data at rest can decrease the risks associated to data disclosures in such scenario. Our proposed solution fits in well with the outsourcing of Big Data processing to the cloud, since information can be stored in encrypted form in external servers in the cloud and processed only if access has been delegated.

Proposal

DASHR: Delegated Access System for Hadoop based on Re-Encryption

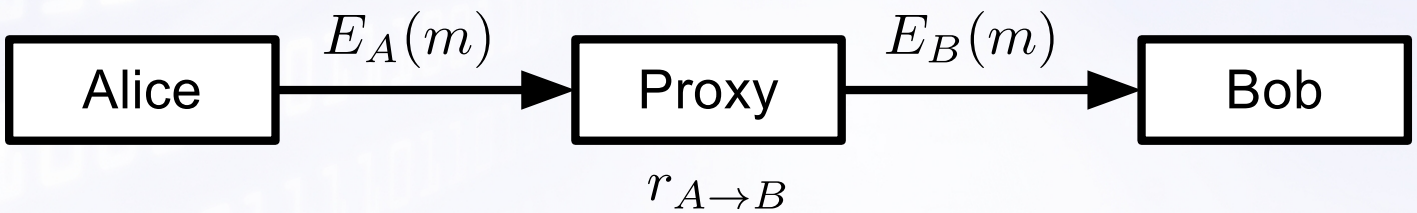
- Cryptographically-enforced access control system
- Based on Proxy Re-Encryption
- Data remains encrypted in the filesystem until it is needed for processing
- Experimental results show that the overhead is reasonable

Hadoop operation



Proxy Re-Encryption: Overview

- A Proxy Re-Encryption scheme is a public-key encryption scheme that permits a proxy to transform ciphertexts under Alice's public key into ciphertexts under Bob's public key.
- The proxy needs a re-encryption key $r_{A \to B}$ to make this transformation possible, generated by the delegating entity
- Proxy Re-Encryption enables delegation of decryption rights

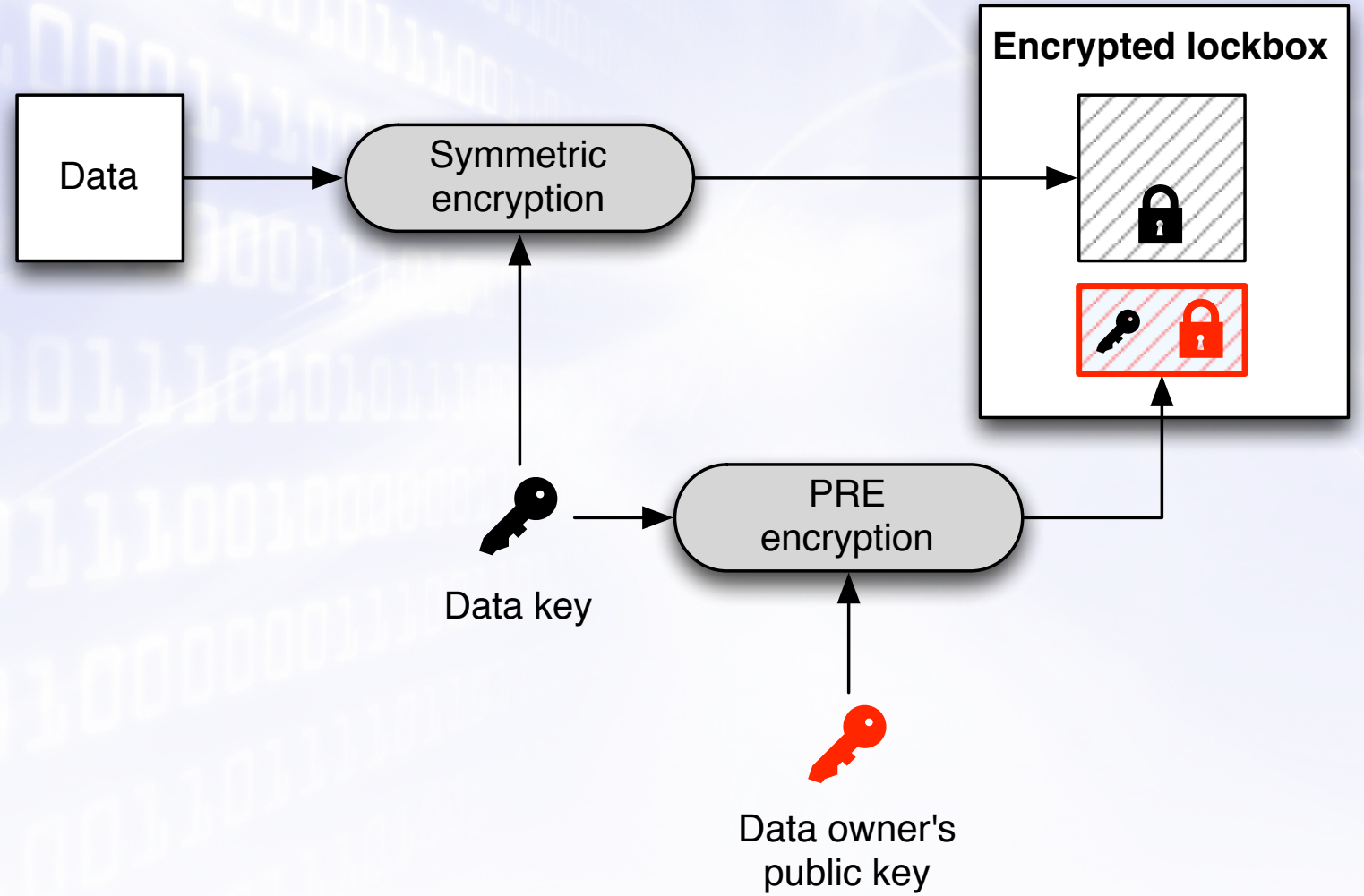


Access Control based on Proxy Re-Encryption

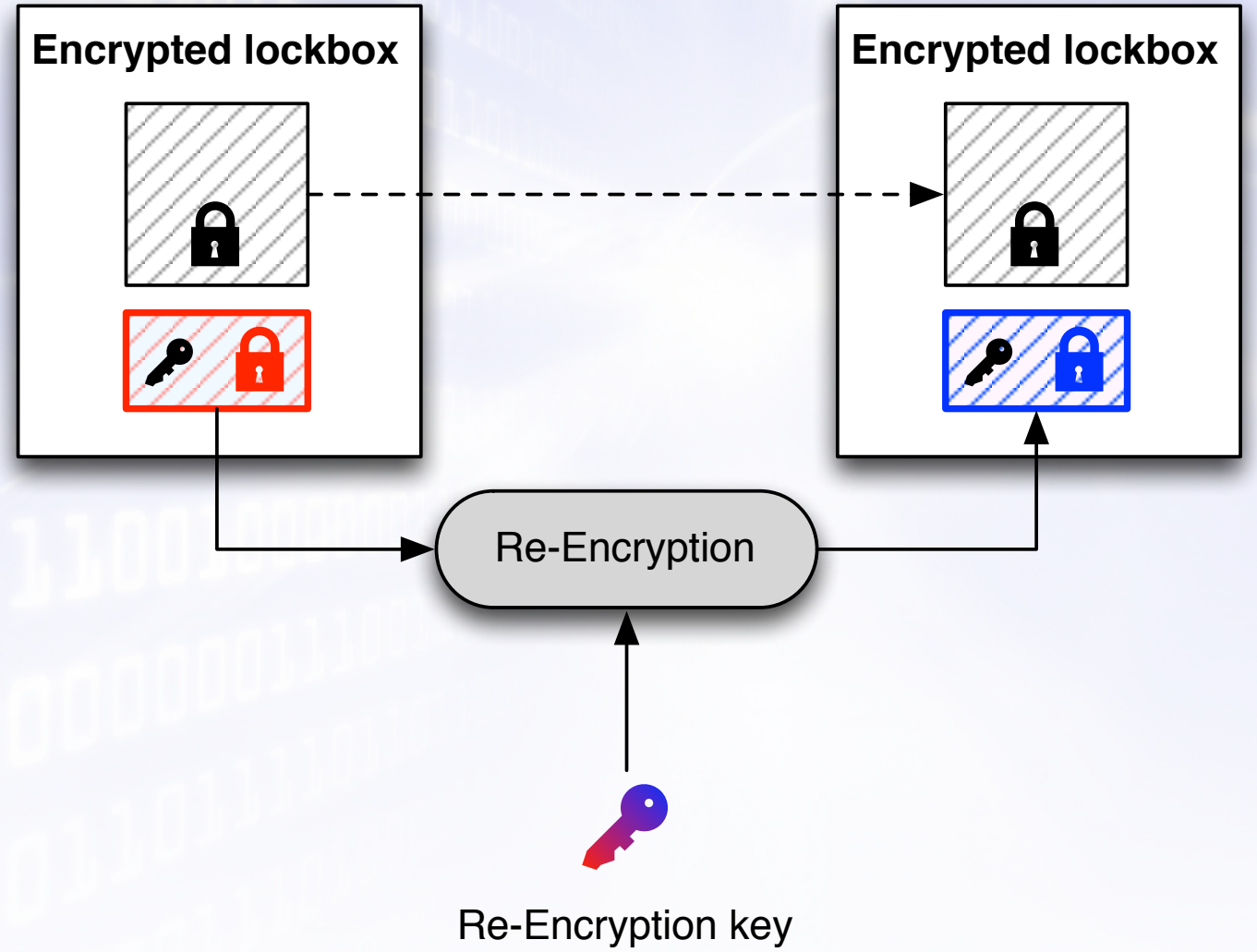
Three entities:

- The data owner, with public and private keys
- The delegatee, with public and private keys
- The proxy entity, which permits access through re-encryption

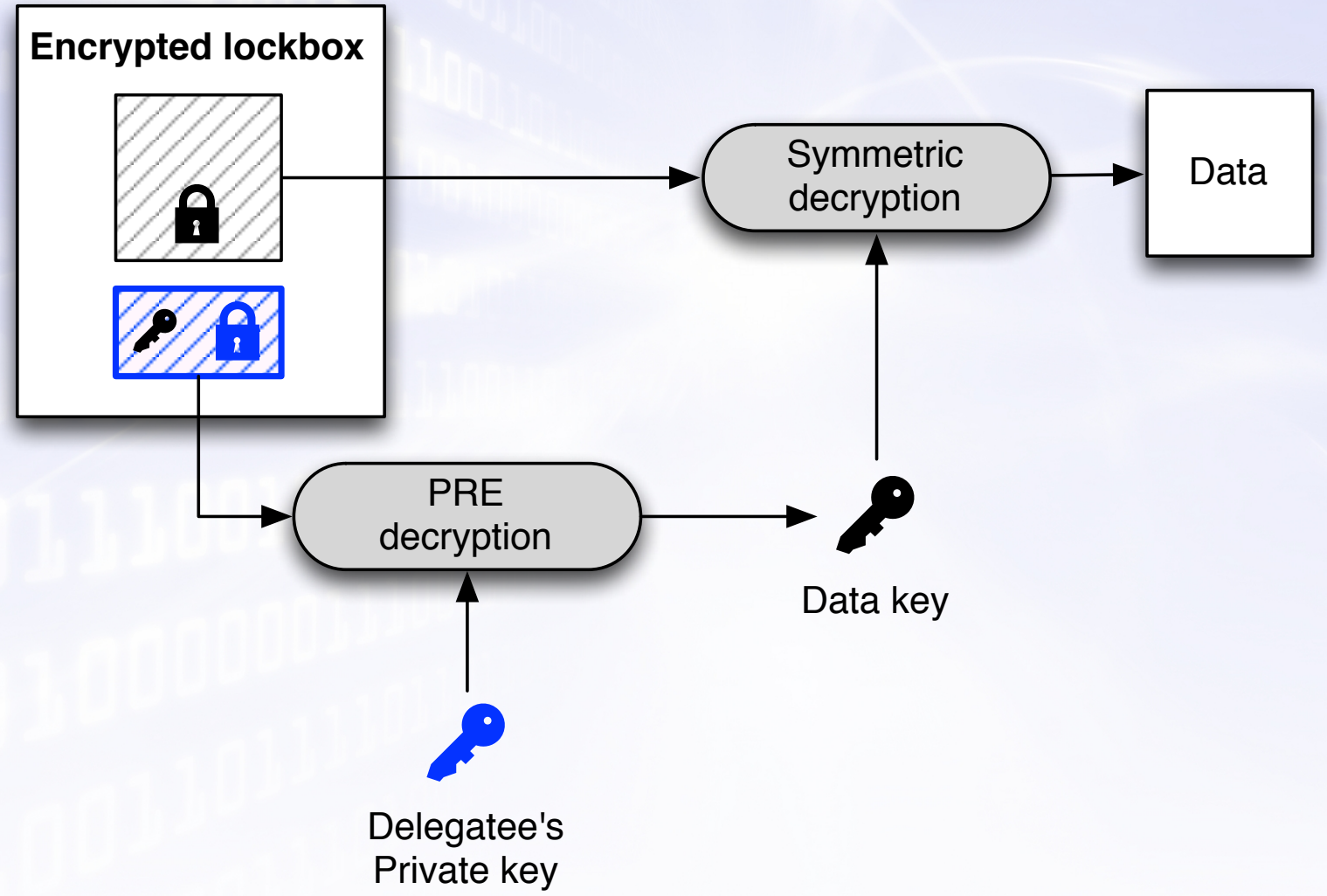
Creating an Encrypted Lockbox



Delegating an Encrypted Lockbox



Opening an Encrypted Lockbox



DASHR

- DASHR: Delegated Access System for Hadoop based on Re-Encryption
- Data is stored encrypted in the cluster and the owner can delegate access rights to the computing cluster for processing.
- The data lifecycle is composed of three phases:
  1. Production phase: during this phase, data is generated by different data sources, and stored encrypted under the owner's public key for later processing.
  2. Delegation phase: the data owner produces the necessary master re-encryption key for initiating the delegation process.
  3. Consumption phase: This phase occurs each time a user of the Hadoop cluster submits a job; is in this phase where encrypted data is read by the worker nodes of the cluster. At the beginning of this phase, re-encryption keys for each job are generated.

DASHR: Production phase

- Generation of data by different sources
- Data is splitted into blocks by the filesystem (e.g., HDFS)
- Each block is an encrypted lockbox, which contains encrypted data and an encrypted key, using the public key of the data owner pk_{DO}

DASHR: Delegation phase

- The dataset owner produces a master re-encryption key mrk_{DO} to allow the delegation of access to the encrypted data
- The master re-encryption key is used to derive re-encryption keys in the next phase.
- The delegation phase is done only once for each computing cluster

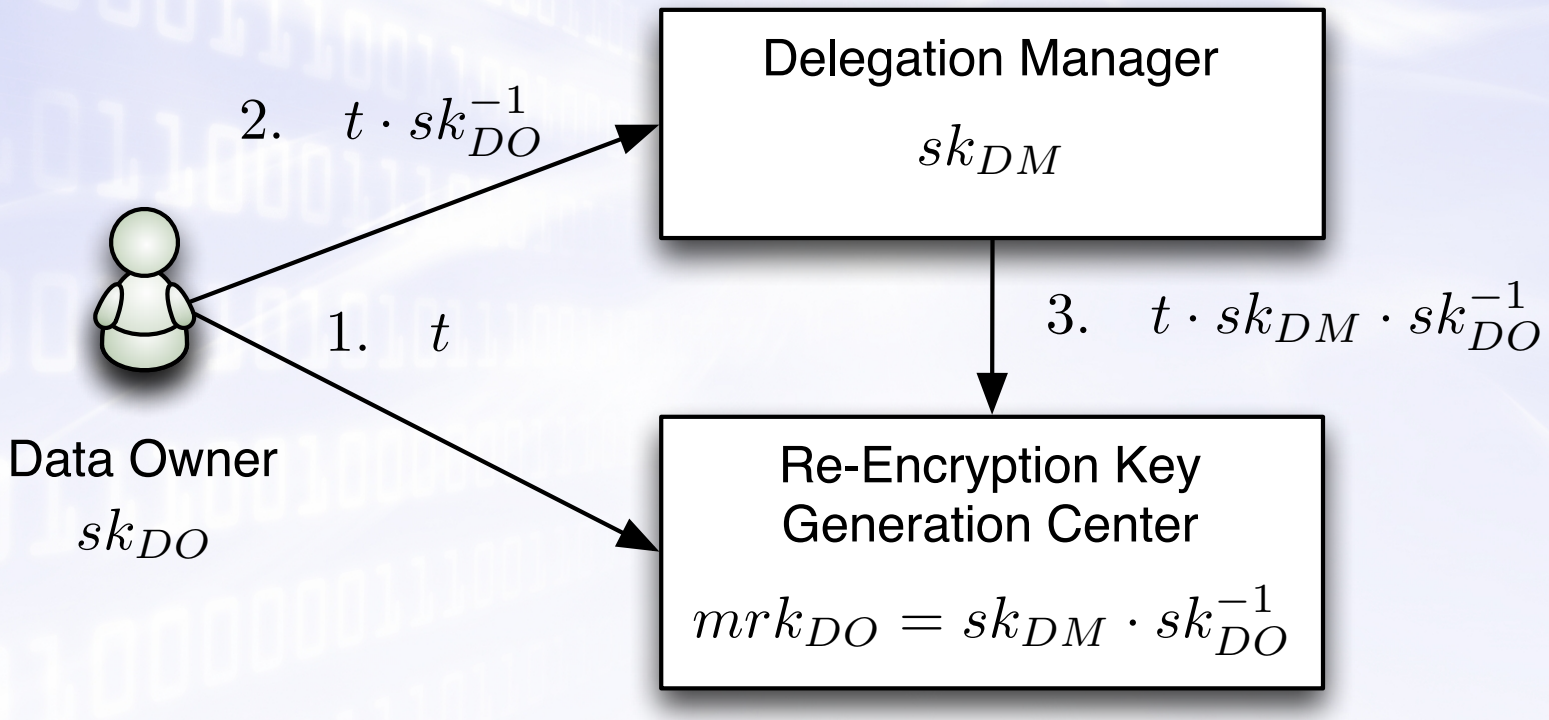
DASHR: Delegation phase

- This phase involves the interaction of three entities:
  1. Dataset Owner (DO), with a pair of public and secret keys (pk_{DO}, sk_{DO}), the former used to encrypted generated data for consumption
  2. Delegation Manager (DM), with keys (pk_{DM}, sk_{DM}), and which belongs to the security domain of the data owner, so it is assumed trusted. It can be either local or external to the computing cluster. If it is external, then the data owner can control the issuing of re-encryption keys during the consumption phase. The delegation manager has a pair of public and secret keys, pk_{DM} and sk_{DM}.
  3. Re-Encryption Key Generation Center (RKGC), which is local to the cluster and is responsible for generating all the re-encryption keys needed for access delegation during the consumption phase.

DASHR: Delegation phase

- These three entities follow a simple three-party protocol, so no secret keys are shared
- The value t used during this protocol is simply a random value that is used to blind the secret key. At the end of this protocol, the RKGC possesses the master re-encryption key mrk_{DO} that later will be used for generating the rest of re-encryption keys in the consumption phase, making use of the transitive property of the proxy re-encryption scheme.

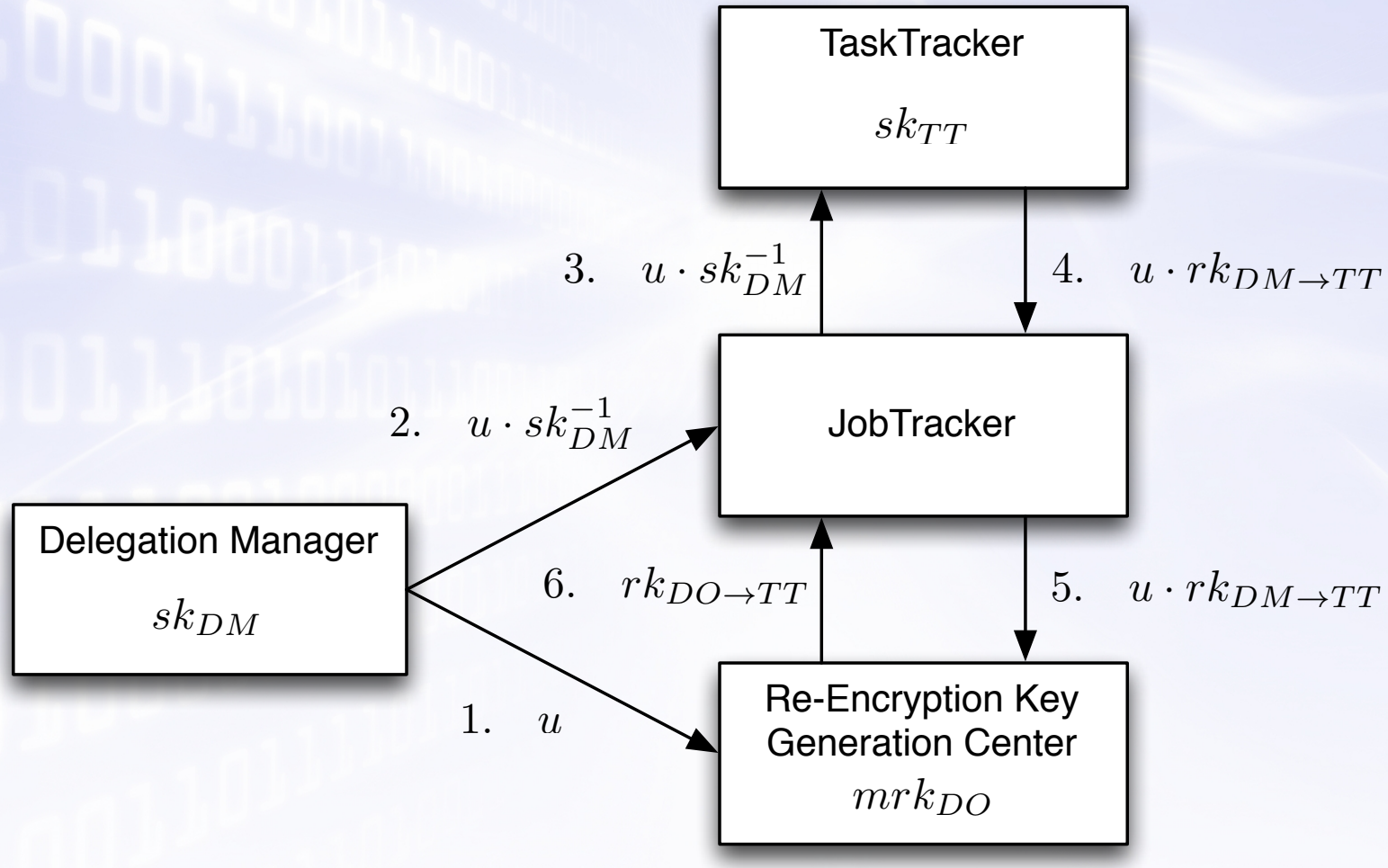
DASHR: Delegation Protocol



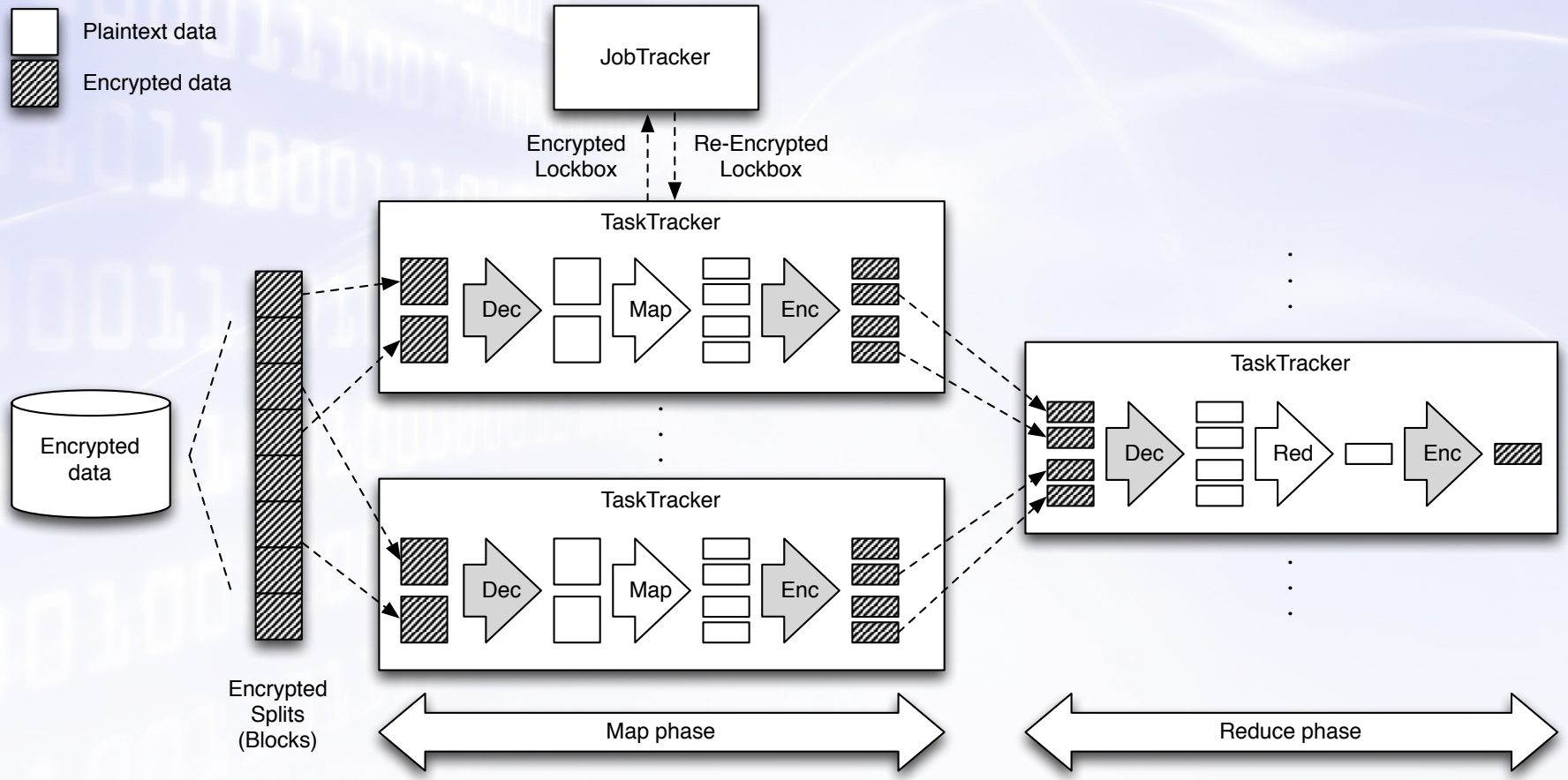
DASHR: Consumption phase

- This phase is performed each time a user submits a job to the Hadoop cluster
- A pair of public and private keys (pk_{TT}, sk_{TT}) for the TaskTrackers is initialized in this step, which will be used later during encryption and decryption.
- The Delegation Manager, the Re-Encryption Key Generation Center, the JobTracker and the TaskTrackers interact in order to generate the re-encryption key $rk_{DO \rightarrow TT}$
- The final re-encryption key $rk_{DO \rightarrow TT}$ held by the JobTracker, who will be the one performing re-encryptions. This process could be repeated in case that more TaskTrackers' keys are in place.

DASHR: Re-Encryption Key Generation Protocol



DASHR: Consumption phase



Experimental setting

- Focused on the main part of the consumption phase, where the processing of the data occurs.
- From the Hadoop perspective, the other phases are offline processes, since are not related with Hadoop's flow.
- Environment:
  - Virtualized environment on a rack of IBM BladeCenter HS23 servers connected through 10 gigabit Ethernet, running VMware ESXi 5.1.0.
  - Each of the blade servers is equipped with two quad-core Intel(R) Xeon(R) CPU E5-2680 @ 2.70GHz.
  - Cluster of 17 VMs (1 master node and 16 slave nodes)
  - Each of the VMs has two logical cores and 4 GB of RAM, running a modified version of Hadoop 1.2.1.

Experimental setting: Cryptographic details

- Proxy Re-Encryption scheme from Weng et al.
- Implemented using the NIST P-256 curve, which provides 128 bits of security and is therefore appropriate for encapsulating 128 bits symmetric keys.
- AES-128-CBC for symmetric encryption.
- We make use of the built-in support for AES included in some Intel processors through the AES-NI instruction set.

Experiment

- Execution of the WordCount benchmark, one of the sample programs included in Hadoop
- A simple application that counts the occurence of words over a set of files.
- In the case of our experiment, the job input was a set of 1800 encrypted files of 64 MB each
- In total, the input contains almost 30 billions of words and occupies aproximately 112.5 GB.
- The size of each input file is sligthly below 64 MB, in order to fit HDFS blocks.

Results

- Two runs over the same input:
  - The first one using a clean version of Hadoop
  - The second one using our modified version.
- The total running time of the experiment was 1932.09 and 1960.74 seconds, respectively.
- That is, a difference of 28.74 seconds, which represents a relative overhead of 1.49%.

Time costs

Operation	Time (ms)
Block Encryption (AES-128-CBC, ~64 MB)	212.62
Block Decryption (AES-128-CBC, ~64 MB)	116.81
Lockbox Encryption (PRE scheme)	17.84
Lockbox Re-Encryption (PRE scheme)	17.59
Lockbox Decryption (PRE scheme)	11.66

Conclusions

- DASHR is a cryptographically-enforced access control system for Hadoop, based on proxy re-encryption
- Stored data is always encrypted and encryption keys do not need to be shared between different data sources.
- The use of proxy re-encryption allows to delegate access to the stored data to the computing cluster.
- Experimental results show that the overhead produced by the encryption and decryption operations is manageable
- Our proposal will fit well in applications with an intensive map phase, since then the overhead introduced by the use of cryptography will be reduced.

Thank you!