## **. Overview of HDFS Transparent Data Encryption (TDE)**

HDFS **TDE secures data at rest** using encryption keys, ensuring that sensitive data remains protected from **unauthorized access**.

✔ **Data is encrypted and decrypted transparently**✔ **Utilizes Key Management Server (KMS)** for **secure key handling**✔ **Enforces security policies with Ranger KMS**

## **🔹 2. How HDFS TDE Works**

HDFS TDE operates using **three key types**:

| **Key Type** | **Function** | **Scope** |
| --- | --- | --- |
| **Data Encryption Key (DEK)** | Encrypts & decrypts file data | Per file |
| **Encrypted Data Encryption Key (EDEK)** | Secures the DEK using the EZ key | Per DEK |
| **Encryption Zone Key (EZK)** | Encrypts/decrypts DEKs | Per Encryption Zone |

✔ **HDFS never stores raw DEKs**, ensuring **strong security**.  
✔ **Encryption is enforced at the file system level**, preventing unauthorized OS-level access.

## **🔹 3. Setting Up HDFS Encryption on Isilon**

### **📌 Step 1: Configure Key Management Server (KMS)**

Install **Ranger KMS**:  
bash  
CopyEdit  
sudo yum install ranger-kms -y

**Create an Encryption Key**:  
bash  
CopyEdit  
hadoop key create my-secure-key -provider kms://http@localhost:9292/kms

### **📌 Step 2: Create an HDFS Encryption Zone**

**Create an HDFS directory**:  
bash  
CopyEdit  
hdfs dfs -mkdir /secure\_zone

**Convert the directory into an Encryption Zone**:  
bash  
CopyEdit  
hdfs crypto -createZone -keyName my-secure-key -path /secure\_zone

**Verify Encryption Zones**:  
bash  
CopyEdit  
hdfs crypto -listZones

✔ Encryption Zones ensure **only authorized users** can access encrypted data.

## **🔹 4. Performance Impact of HDFS Encryption**

A benchmark using **Teragen, Terasort, and Teravalidate** was conducted to measure **the impact of TDE on Hadoop jobs**.

| **Test** | **Performance Impact** |
| --- | --- |
| **Teragen** | 8.3% slower |
| **Terasort** | 2.4% slower |
| **Teravalidate** | 5.9% slower |

✔ **Write operations are slower (~8-10%)** due to encryption overhead.  
✔ **Read operations are slower (~5-7%)** due to decryption processing.  
✔ **AES-NI Hardware Acceleration** can significantly reduce the performance impact.

## **🔹 5. Validating Encryption with Ranger KMS**

**Test Writing Encrypted Data**bash  
CopyEdit  
echo "Sensitive Data" > confidential.txt

hdfs dfs -put confidential.txt /secure\_zone/

**Verify Data Encryption**bash  
CopyEdit  
hdfs dfs -ls /secure\_zone/

1. ✅ The file should be listed **normally**.

**Attempt Unauthorized Access**bash  
CopyEdit  
sudo -u unauthorized\_user hdfs dfs -cat /secure\_zone/confidential.txt

1. ❌ Expected Output: **Permission Denied**

✔ Ranger KMS **enforces strict encryption policies** to prevent unauthorized access.

## **🔹 6. Audit Logging & Monitoring**

To enable **audit logging** for Ranger KMS:

**Configure Ranger KMS Audit Logs**:  
bash  
CopyEdit  
sudo nano /etc/ranger/kms/conf/kms-site.xml

Set **HDFS Audit Destination**:  
xml  
CopyEdit  
<property>

<name>xasecure.audit.destination.hdfs</name>

<value>true</value>

</property>

**Restart Ranger KMS**:  
bash  
CopyEdit  
sudo systemctl restart ranger-kms

✔ **All key access and encryption operations** are logged for **compliance and security monitoring**.

## **🔹 7. Summary**

| **Feature** | **Benefit** |
| --- | --- |
| **HDFS Encryption Zones** | Secures data at **file system level** |
| **Key Management with KMS** | Prevents **unauthorized key access** |
| **Ranger KMS Policy Enforcement** | Ensures **fine-grained access control** |
| **Performance Impact (~5-10%)** | Manageable with **AES-NI optimizations** |
| **Audit Logging** | Tracks **encryption and access events** |

✔ **HDFS Transparent Data Encryption (TDE) is essential for protecting sensitive data in Hadoop environments**.  
✔ **Implementing Ranger KMS ensures centralized and secure key management**.  
✔ **Optimizations like AES-NI can significantly reduce encryption overhead**.