**Background**

The protection of sensitive data stored within organizational on-premises systems is critical to ensuring confidentiality, integrity, and compliance with regulatory standards. Challenges in implementing data encryption at rest have resulted in vulnerabilities and audit findings:

* Sensitive data stored without encryption in on-premises production and non-production environments is susceptible to unauthorized access.
* Inconsistent encryption practices within on-premises data centers increase the risk of data breaches.
* Poor management of cryptographic keys compromises the security of encrypted data.

*Note: This implementation is tailored for on-premises environments, specifically GS data centers, and does not address cloud-specific deployments unless explicitly integrated with on-prem systems.*

This document proposes a reference implementation for data encryption at rest to mitigate these risks, safeguard sensitive data, and support operational and development needs while aligning with firmwide policies and standards such as DP30\*\*.

**Definitions**

* **Sensitive Data** - Data classified as DP20 or above per the Firmwide Policy for Data Classification, including personally identifiable information (PII), financial records, and proprietary data.
* **Encryption at Rest** - The cryptographic protection of data stored on disks, databases, or other media to prevent unauthorized access, as mandated by the firm’s cryptography standards.
* **Environment** - A segregated set of computing resources designated for a specific purpose within on-premises infrastructure.
* **Production** - Current classification for on-premises environments that support the firm’s business processes, where infrastructure, systems, and applications conduct, process, or record firm business.
* **Non-Production** - Current classification for on-premises environments used for development and testing purposes.
* **Secure Environment** - Subject to the control requirements and governance mandated for handling sensitive data within on-premises systems.
* **Data Read Sensitive Environment** - Contains or processes data that is not publicly available within on-premises setups.
* **Data Write Sensitive Environment** - Contains or modifies data whose correctness must be guaranteed by modification through approved processes within on-premises systems.
* **Operationally Sensitive Environment** - Critical to end-user activities within on-premises infrastructure, where failure would disrupt business or testing operations.
* **Business Sensitive Environment** - Produces output directly used in business processes within on-premises environments.

**Considerations when working with data encryption at rest**

* Sensitive data must be protected from unauthorized access when stored, aligning with the firm’s cryptography standards.
* Operational integrity of on-premises production environments must be maintained during encryption processes.
* Productivity of developers, strategists, and business users requiring access to encrypted data must be preserved.

**Risks and Threat Vectors**

| **Class** | **Risk** | **Threat Vector** | **Examples** |
| --- | --- | --- | --- |
| Data | Unauthorized Access | Physical theft of unencrypted storage media | - A stolen unencrypted hard drive from a data center exposes customer PII and financial data. |
|  |  | Insider extraction of unencrypted data | - A privileged user copies unencrypted database backups to an external device. |
|  | Key Compromise | Exposure of cryptographic keys | - Keys stored in a misconfigured repository are accessed by an unauthorized user. |
|  |  | Insufficient key rotation | - Stale keys are exploited to decrypt archived sensitive data. |
|  | Data Leakage | Unencrypted data transfer | - Unencrypted backups are copied to an insecure on-premises server. |
|  |  | Misconfigured encryption settings | - A database is deployed without enabling encryption at rest, exposing sensitive records. |
|  | Data Loss | Loss of encryption keys | - Inaccessible keys render encrypted data unrecoverable, disrupting business operations. |
| Operational | Resource Starvation | Encryption processing overhead | - CPU-intensive encryption delays transaction processing, impacting service-level objectives (SLOs). |
|  | System Downtime | Key management system failure | - A corrupted key vault prevents data decryption, halting critical applications. |
|  |  | Encryption misconfiguration | - Incorrect encryption settings cause application failures, leading to downtime. |

**Areas Requiring Further Review**

* Evaluation of legacy on-premises systems’ compatibility with AES-256 encryption.
* Assessment of on-premises key management integration with existing HSMs.
* Validation of performance impacts on high-traffic on-premises production environments.
* Review of developer access policies for encrypted test datasets to ensure compliance with entitlement controls (e.g., AC-3.106).

**Pending Approvals**

* Approval from Business Sponsors for encryption workflow implementation (e.g., AC-3.105).
* Sign-off by Compliance Team on audit trail integration with on-premises SIEM systems.
* Authorization from Privilege Managers for RBAC configurations in non-production environments.

**Requirements**

* Sensitive data when stored or accessed must be encrypted at rest using approved cryptographic keys and mechanisms that comply with the firm’s cryptography standard, as per DP30\*\*.
* Encryption keys must be managed in a secure on-premises system with access controls, rotation policies, and audit logging.
* Data used for business processes must not be accessible in unencrypted form outside of approved on-premises processes.
* Encryption processes must not compromise the operational integrity or performance of on-premises production environments.
* Developers must be able to perform testing with encrypted data that complies with the firm’s mandated data access controls, using techniques like masking or tokenization where necessary.

**Proposal**

| **Feature** | **Immediate** | **Short Term** | **Target** |
| --- | --- | --- | --- |
| Identification of encryption coverage | - Storage will be assessed to ensure that all audit and security controls for sensitive data are classified | - Add questions around operational, purpose, and data sensitivity to ARP | - Add operational, purpose, and data sensitivity to AppDir |
| Separation of Production - Non-Production Encryption | - Validation: App Team to document multi-environment architecture as deployment diagram, to be reviewed by xox (?) prior to onboarding | - Validation: App Team to document multi-environment architecture as deployment diagram, to be reviewed by xox (?) prior to onboarding | - Ensure 100% separation of encryption keys between production and non-production |
| Key Management Security | - Remote processes run with system account credentials | - Remote processes run with system account credentials | - Static analysis in SDLC pipeline to detect unauthorized key access |
| Performance Optimization | - Only run remote processes from controlled code bases (protected branches) | - Implement hardware-accelerated encryption (e.g., Intel AES-NI) | - Achieve <5% latency increase across all on-premises environments |

**Features and Intermediate Short-Term Targets**

* **Features**:
  + Automated key rotation and management via a centralized on-premises KMS with HSM integration.
  + Hardware-accelerated encryption (e.g., Intel AES-NI) to minimize performance impact.
  + Real-time audit logging of encryption and decryption events, compliant with AU-3.102.
  + Encrypted test datasets with masking for on-premises non-production environments, supporting developer productivity.
* **Intermediate Short-Term Targets**:
  + Month 2: Complete KMS pilot deployment with initial key rotation (Infrastructure Team).
  + Month 4: Achieve 50% encryption coverage in on-premises production databases (Database Team).
  + Month 6: Extend encryption to 75% of on-premises non-production environments (DevOps Team).
  + Month 7: Validate backup encryption and audit trail integration (Backup Team).

**Reference Implementation Proposal**

To address the identified risks and meet the requirements, the following reference implementation for data encryption at rest is proposed:

1. *Encryption Standards*
   * Adopt AES-256 encryption in Galois/Counter Mode (GCM) for all sensitive data at rest to ensure confidentiality and integrity, compliant with FIPS 140-3 and the firm’s cryptography standard.
   * Use cryptographic modules validated by Data Asset Owners to protect sensitive data residing in on-premises GS data centers.
2. *Key Management*
   * Deploy a centralized on-premises Key Management System (KMS) to generate, store, and manage encryption keys securely, leveraging Hardware Security Modules (HSMs) for key generation.
   * Implement automated key rotation every 12 months, with versioning to prevent data loss, and enforce role-based access controls (RBAC) as defined by Entitlement Approvers and Administrators.
   * Maintain audit trails of key operations, aligning with control specifications (e.g., AU-3.102).
3. *Environment-Specific Encryption*
   * *Production*: Enable encryption at rest for all databases (e.g., SQL Server Transparent Data Encryption), file systems (e.g., dm-crypt), validated by Technical Owners.
   * *Non-Production*: Encrypt sensitive data with separate keys to limit risk exposure, ensuring compliance with production-level controls and approved entitlement workflows (e.g., AC-3.105).
   * *Backups*: Encrypt all backup data using on-premises KMS-managed keys, with access restricted to authorized recovery processes.
4. *Performance Optimization*
   * Leverage hardware-accelerated encryption (e.g., Intel AES-NI) to minimize CPU overhead and maintain performance, monitored by Solution Owners.
   * Configure encryption at the storage layer to reduce application-level changes, ensuring operational integrity.
5. *Access and Audit Controls*
   * Implement RBAC to restrict decryption capabilities to authorized users and services, with Privilege Managers reviewing entitlements quarterly (e.g., AC-6.7.100).
   * Log all encryption and decryption operations in a tamper-proof audit trail, integrated with the firm’s on-premises central inventory (e.g., AC-24.1).
   * Conduct quarterly validations of encryption feeds to the central inventory, as required by Technical Owners (e.g., AC-24.1.100).
6. *Development and Testing*
   * Provide developers with encrypted test datasets using masked or synthetic data, compliant with data access controls and entitlement management systems (e.g., PERMIT Central).
   * Use on-premises sandboxed environments with KMS-managed keys for testing encryption workflows, ensuring no self-approval (e.g., AC-3.106).

**Implementation Roadmap**

| **Phase** | **Activity** | **Timeline** | **Owner** |
| --- | --- | --- | --- |
| 1 | Assess current encryption posture and gaps | Month 1 | Security Team |
| 2 | Deploy KMS and integrate HSMs | Months 2-3 | Infrastructure Team |
| 3 | Enable encryption for production databases and storage | Months 4-5 | Database Team |
| 4 | Extend encryption to non-production environments | Month 6 | DevOps Team |
| 5 | Implement backup encryption and key rotation policies | Month 7 | Backup Team |
| 6 | Conduct compliance audit and address findings | Month 8 | Compliance Team |

**Metrics for Success**

* Percentage of sensitive data encrypted at rest: Target 100%.
* Number of key compromise incidents: Target 0.
* Performance impact from encryption: Target <5% latency increase.
* Compliance audit findings related to encryption: Target 0.

**Risk Mitigation**

* *Key Loss*: Implement redundant on-premises KMS instances and maintain secure key backups.
* *Performance Degradation*: Optimize encryption with hardware acceleration and monitor resource usage.
* *Unauthorized Access*: Enforce least privilege access and monitor key usage for anomalies, with Privilege Manager oversight.
* *Non-Compliance*: Conduct regular audits to verify adherence to DP30\*\* and internal standards.

**Environment Encryption Matrix**

| **Environment** | **Data Read Sensitive** | **Data Write Sensitive** | **Operational Sensitivity** | **Business Sensitivity** | **Encryption Required** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Production Database | Y | Y | Y | Y | Y | AES-256 with KMS-managed keys |
| Non-Production Database | Y | N | N | N | Y | Separate keys from production |
| File Storage | Y | Y | N | Y | Y | Disk-level encryption (e.g., dm-crypt) |
| Backups | Y | N | N | Y | Y | KMS-encrypted archives |
| Development Sandbox | N | N | N | N | N | Use tokenized or synthetic data |

**Encryption Configuration Matrix**

| **Data Read** | **Data Write** | **Operational** | **Business** | **End User** | **Environment** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Y | Y | Y | Y | Application | Production Database | TDE with AES-256, KMS keys |
| Y | N | N | N | Developer | Non-Production Database | TDE with separate KMS keys |
| Y | Y | N | Y | Application | File Storage | dm-crypt with AES-256 |
| Y | N | N | Y | Admin | Backups | KMS-encrypted archives |
| N | N | N | N | Developer | Sandbox | No encryption; synthetic data |

**Conclusion**

Implementing data encryption at rest is essential to protect sensitive data, ensure compliance with standards like DP30\*\*, and mitigate risks of unauthorized access or data loss within on-premises environments. This reference implementation, supported by the proposal, features, and targets, provides a comprehensive framework to encrypt data, manage keys securely, and support development needs without compromising performance or security. By adopting these controls, the organization can strengthen its data protection posture and align with industry and regulatory requirements.