CA Key Lifecycle and Termination Policy

DdSign Electronic Certification Service Provider

Private Key Generation, Protection, Backup, Termination, and Archival Procedures

Document Reference	CENT-ECSP-KLP-001
Version	1.0
Classification	Strictly Confidential - Internal & Auditor Use
Owner	Centric Limited
Author	Arnold Wamae
Effective Date	15 April 2026
Review Date	Annually, or upon any key lifecycle event

Document Control

Version History

Version	Date	Author	Change Summary	Approved By
1.0	15 April 2026	Arnold Wamae	Initial release for E-CSP audit	Margret Mumbi

Approval

Role	Name	Signature	Date
Author	Arnold Wamae		15 April 2026
Reviewer	Arnold Wamae (ID 3523005)		15 April 2026
Approver	Margret Mumbi, MD, Innovation		

1. Purpose and Scope

1.1 Purpose

This document defines the policy and procedures governing the complete lifecycle of the ddsign E-CSP's Certification Authority (CA) signature private keys, from generation through operational use to termination and archival. It ensures that upon termination of a CA signature private key, encrypted backup copies are securely created, archived, and stored in accordance with the regulatory requirements for E-CSPs in Kenya.

This policy directly addresses the auditor requirement that the E-CSP archives and securely stores backup copies upon the termination of the E-CSP's signature private key.

1.2 Scope

This policy applies to:

- The ddsign Issuing CA private key (RSA 4096-bit), used to sign Subscriber certificates and CRLs.
- Any future CA keys generated by the E-CSP, including replacement keys generated after a key termination event.
- Subscriber private keys generated and held within the cert-management system.
- All DDSign Operations staff involved in key generation, key ceremony, key backup, and key termination procedures.

1.3 Key Storage Modes

The cert-management system supports two key storage modes, each with distinct lifecycle procedures:

Mode	Implementation	Key Location	Backup Method	Termination Method
Software (SoftwareKeyStore)	AES-256-GCM encrypted files on disk. Key Encryption Key (KEK) loaded from environment variable at runtime.	Encrypted .enc files in the configured key directory (HSM_KEY_PATH).	BackupKey(): exports key encrypted with AES-256-GCM using a passphrase-derived key (scrypt, N=32768, r=8, p=1).	DestroyKey(): overwrites key file with cryptographically random data, then deletes the file.
CloudHSM (PKCS#11)	AWS CloudHSM via PKCS#11 (crypto11 library). Private key never leaves HSM hardware.	Inside the HSM hardware. Not extractable.	HSM native backup mechanism (AWS CloudHSM cluster cloning / key wrapping).	PKCS#11 key object deletion via HSM API.

1.4 Regulatory Basis

- **KICA, CAP 411A:** Requires E-CSPs to protect CA private keys and maintain secure backup and recovery capabilities.
- **ETSI EN 319 411-1:** Requires that private keys are securely archived upon end of life.
- **CPS Section 5.7:** Mandates compromise and disaster recovery procedures for the ddsign CA, including key backup.

- **CPS Section 6.2:** Defines private key protection and cryptographic module engineering controls.

2. Key Lifecycle Overview

The CA signature private key passes through the following lifecycle phases:

Phase	Description	Authorisation Required
1. Generation	A new RSA key pair is generated inside the key store (software or HSM) during a witnessed key ceremony.	Dual authorisation: Technical Lead + Incident Commander
2. Activation	The CA certificate is issued (self-signed for development, or cross-signed by KE-RCA for production). The key becomes operational.	Technical Lead (self-sign) or KE-RCA (cross-sign)
3. Operational Use	The key signs Subscriber certificates and CRLs during normal E-CSP operations.	System (automated per certificate request)
4. Suspension (if needed)	The key is temporarily removed from operational use during an investigation. No certificates are signed. Certificate status services continue.	Dual authorisation: Technical Lead + Incident Commander
5. Termination	The key is permanently retired. A final CRL is signed. Encrypted backups are created. The operational copy is securely destroyed.	Dual authorisation: Technical Lead + Incident Commander
6. Archival	Encrypted backups are transferred to secure archive storage locations. The key exists only in encrypted archive form.	Technical Lead confirms archive; Incident Commander approves
7. Archive Disposal	After the retention period expires, archived backups are securely destroyed.	Dual authorisation: Technical Lead + Incident Commander

3. Key Generation

3.1 Key Generation Procedure

The CA key pair is generated during a witnessed key ceremony:

1. **Authorisation:** The Technical Lead and the Incident Commander both authorise the key generation.
2. **Environment:** The ceremony is conducted in a physically secure environment. Only authorised participants are present.
3. **Execution:** The key generation command is executed on the cert-management system. For software mode, this calls KeyStore.GenerateKeyPair() which creates an RSA key pair and encrypts it with AES-256-GCM under the KEK before writing to disk. For CloudHSM mode, the key is generated inside the HSM hardware via PKCS#11.
4. **Key parameters:** The CA key size is 4096 bits (RSA), as configured in the CA_KEY_SIZE environment variable.
5. **Witness record:** Each participant signs the Key Ceremony Record, confirming their presence, the key label generated, and the timestamp.
6. **Audit log:** An audit trail entry is created in the system (action: ca.key.generated) recording the actor, timestamp, and IP address.

3.2 Key Labels and Identification

Each key is identified by a unique label:

- **Issuing CA key:** Label: 'issuing-ca'. Stored as issuing-ca.enc (software) or identified by PKCS#11 label (CloudHSM).
- **Subscriber keys:** Label: 'subscriber-{serial}' where {serial} is the 128-bit certificate serial number in hexadecimal.

4. Key Protection During Operational Use

4.1 Software Key Store Protection

- **Encryption at rest:** All private keys are encrypted with AES-256-GCM using a Key Encryption Key (KEK). The encrypted file format is: base64(nonce[12] + ciphertext + GCM-tag[16]). Files use the .enc extension.
- **KEK management:** The KEK is a 32-byte (256-bit) key loaded from the KEY_ENCRYPTION_KEY environment variable at runtime. It is never written to disk. If the KEK is not configured, the system falls back to plaintext PEM storage (development mode only; not permitted in production).
- **File permissions:** Key files are written with 0600 permissions (owner read/write only). The key directory is created with 0700 permissions.
- **Access control:** Access to the key directory and the KEK environment variable is restricted to the Technical Lead and the CA Operations Lead, as defined in the Access Control Matrix (CENT-ECSP-ACM-001).

4.2 CloudHSM Protection

- **Hardware boundary:** The CA private key never leaves the HSM hardware. All signing operations occur inside the HSM.
- **Authentication:** Access to the HSM requires a Crypto User (CU) PIN, configured via the HSM_PIN environment variable.
- **Tamper protection:** AWS CloudHSM provides FIPS 140-2 Level 3 validated hardware with tamper-detection and response mechanisms.
- **Audit:** The HSM logs all key usage events independently of the application audit trail.

4.3 Operational Monitoring

- The cert-management system audit trail (INSERT-only, no UPDATE or DELETE) records every certificate issuance, revocation, suspension, and reactivation event with the actor, target serial, details, IP address, and timestamp.
- The Sentry monitoring platform (sentry.centricltd.co.ke) monitors the cert-management application for errors and anomalies.
- Key usage frequency is reviewable via the audit trail and the dashboard statistics endpoint.

5. Key Backup During Operational Life

5.1 Routine Backup (Software Mode)

Routine backups of the CA private key are created periodically and after significant events (e.g., after initial key generation, after a large batch of certificate issuances).

7. **Execution:** The Technical Lead initiates the backup by calling BackupKey() with a strong passphrase (minimum 32 characters, generated from a cryptographically secure random source).
8. **Encryption:** BackupKey() loads and decrypts the key from disk (using the KEK), then re-encrypts the raw key material with AES-256-GCM using a key derived from the passphrase via scrypt (N=32768, r=8, p=1, key length=32 bytes). A random 32-byte salt is generated for each backup.
9. **Output format:** The output format is: salt[32] + nonce[12] + ciphertext + GCM-tag[16]. This is a self-contained encrypted blob that can be decrypted with the passphrase alone (no dependency on the KEK).
10. **Storage:** The encrypted backup is stored in two geographically separate secure locations (see Section 5.3).
11. **Passphrase custody:** The passphrase is recorded on paper, placed in a tamper-evident envelope, and stored in a physical safe accessible only to the Technical Lead and the Incident Commander. The passphrase is never stored digitally.

5.2 Routine Backup (CloudHSM Mode)

For CloudHSM deployments, routine backup uses the HSM's native mechanisms:

- AWS CloudHSM automatically maintains encrypted backups of all keys across the HSM cluster nodes.
- For off-site archival, the HSM cluster backup is initiated via the AWS CloudHSM management API.
- The software BackupKey() method is not supported for CloudHSM keys (the private key is non-extractable).

5.3 Backup Storage Locations

Location	Type	Access Control	Description
Primary	Secure cloud storage	Restricted to Technical Lead and Incident Commander. Encrypted at rest (storage-level encryption in addition to the backup's own AES-256-GCM encryption).	The primary backup copy, stored in a secure cloud storage bucket with versioning enabled and access logging.
Secondary	Physical off-site safe	Physical key held by the Incident Commander. Tamper-evident packaging.	An offline copy on encrypted removable media, stored in a fireproof safe at a separate physical location from the primary data centre / office.

5.4 Backup Verification

After each backup:

- The Technical Lead verifies that the encrypted backup file is non-zero in size and can be read from each storage location.
- A test decryption is performed on a copy of the backup (in a secure, isolated environment) to confirm the passphrase produces valid key material. The test copy is securely deleted immediately after verification.
- The backup event is recorded in the audit trail (action: ca.key.backup.created) and in the Key Lifecycle Log (Section 10).

6. Conditions for Key Termination

The CA signature private key shall be terminated under any of the following conditions:

Condition	Description	Urgency
Scheduled key rollover	The CA key has reached its planned end-of-life (typically after the CA certificate's validity period, or as determined by the key rollover schedule).	Planned. Termination scheduled at least 30 days in advance.
CA key compromise	The CA private key is confirmed or strongly suspected to be compromised. This is a Critical (S1) incident per the Incident Response Action Plan.	Immediate. Termination executed as part of the incident response.
Regulatory directive	The Communications Authority of Kenya or KE-RCA directs the E-CSP to terminate the CA key.	As directed. Typically immediate or within 24 hours.
E-CSP cessation of operations	Centric Limited ceases to operate the ddsign E-CSP.	Planned. Termination as part of the wind-down procedure.
Cryptographic algorithm deprecation	The key algorithm (RSA) or key size (4096-bit) is deemed insufficient by the relevant standards body or regulatory authority.	Planned. Transition to a new key with the replacement algorithm.
HSM end-of-life	The hardware security module reaches its end of life or is decommissioned.	Planned. Key migrated to replacement HSM before termination.

7. Key Termination Procedure

The following procedure is executed when a CA signature private key is terminated. All steps are performed in sequence. The procedure requires dual authorisation from the Technical Lead and the Incident Commander.

7.1 Pre-Termination Checks

12. **Dual authorisation:** The Technical Lead and the Incident Commander both authorise the key termination in writing (email or signed form). Neither party may authorise alone.
13. **Certificate inventory:** The Technical Lead confirms that all Subscriber certificates issued under the key being terminated are accounted for. A complete list of active, suspended, and revoked certificates is exported from the cert-management database.
14. **Replacement key ready:** If the termination is planned (not a compromise), the Technical Lead verifies that a replacement CA key has been generated and activated, and that new Subscriber certificates have been (or will be) issued under the replacement key.
15. **Incident response coordination:** For compromise scenarios, the Incident Response Action Plan (CENT-ECSP-IRP-001) is activated in parallel. The key termination procedure below is executed as part of the incident response.

7.2 Final CRL Issuance

16. **Generate final CRL:** The Technical Lead initiates the generation of a final Certificate Revocation List (CRL) signed by the key being terminated.
17. **CRL content:** The final CRL includes all certificates revoked under this CA key, including any mass revocations triggered by the termination event.
18. **CRL validity:** The final CRL is published to the CRL distribution point(s) with an extended validity period (recommended: 365 days, or until the last certificate issued under this key expires, whichever is longer).
19. **CRL verification:** The CRL publication is verified: the Technical Lead confirms the CRL is accessible at the distribution point and parses correctly.
20. **Audit log:** An audit trail entry is created (action: ca.crl.final.issued).

7.3 Encrypted Backup Creation

Before the operational key is destroyed, encrypted backup copies are created and verified. This is the critical step that satisfies the auditor requirement.

7.3.1 Software Key Store

21. **Passphrase generation:** The Technical Lead generates a strong passphrase (minimum 32 characters) using a cryptographically secure random generator.
22. **Backup execution:** The Technical Lead calls BackupKey('issuing-ca', passphrase) on the cert-management system. This produces an encrypted blob: salt[32] + nonce[12] + ciphertext + GCM-tag[16], encrypted with AES-256-GCM using a scrypt-derived key.
23. **Output capture:** The encrypted backup blob is written to removable media (e.g., USB drive) in the presence of the Incident Commander.

24. **Second copy:** A second copy of the encrypted backup is uploaded to the primary secure cloud storage location.
25. **Verification:** Both copies are verified by test decryption in an isolated environment. The test environment is securely wiped after verification.
26. **Passphrase custody:** The passphrase is written on paper, placed in a tamper-evident envelope, sealed, signed across the seal by both the Technical Lead and the Incident Commander, and stored in the physical safe at the secondary storage location.
27. **Audit log:** An audit trail entry is created (action: ca.key.backup.termination) recording the timestamp, the actors, and the storage locations.

7.3.2 CloudHSM

28. **HSM backup:** An HSM cluster backup is initiated via the AWS CloudHSM management API.
29. **Verification:** The backup is verified by confirming its presence in the AWS backup inventory.
30. **Record:** The HSM cluster backup identifier, timestamp, and storage region are recorded in the Key Lifecycle Log.
31. **Decommission export:** If the HSM is being decommissioned entirely, a wrapped key export (using the HSM's key wrapping mechanism) is performed before cluster deletion, and the wrapped key blob is stored in the same two locations as described in Section 5.3.

7.4 Operational Key Destruction

Once the encrypted backups are created, verified, and stored, the operational copy of the key is securely destroyed.

7.4.1 Software Key Store

32. **Execute destruction:** The Technical Lead calls DestroyKey('issuing-ca') on the cert-management system.
33. **Secure wipe:** DestroyKey() overwrites the key file with cryptographically random data (matching the file's original size), then deletes the file from disk.
34. **Verification:** The Technical Lead verifies that the key file no longer exists at the configured key path (HSM_KEY_PATH/issuing-ca.enc or HSM_KEY_PATH/issuing-ca.pem).
35. **Witness confirmation:** The Incident Commander confirms the deletion by independent inspection of the key directory.
36. **Audit log:** An audit trail entry is created (action: ca.key.destroyed) recording the timestamp, actors, and confirmation of deletion.

7.4.2 CloudHSM

37. **Execute destruction:** The Technical Lead deletes the key object from the HSM via the PKCS#11 API (DestroyKey()).
38. **Verification:** The deletion is verified by attempting to load the key (LoadKey()), confirming it returns 'key not found'.
39. **HSM audit:** The HSM audit log is reviewed to confirm the deletion event.

7.5 CA Certificate Archival

In addition to the private key backup, the CA certificate (public) is archived:

- The CA certificate PEM file (certs/issuing-ca.pem) is copied to both archive storage locations alongside the encrypted key backup.
- The CA certificate remains publicly available at the CRL distribution point and in the PKI repository for the duration of the archive retention period, so that Relying Parties can continue to validate previously-signed documents.

7.6 Post-Termination Notification

40. **Regulatory notification:** The Communications Authority of Kenya is notified of the key termination within 24 hours (for unplanned terminations) or 30 days in advance (for planned terminations).
41. **KE-RCA notification:** KE-RCA is notified to update the cross-certification status if applicable.
42. **Subscriber notification:** Subscribers are notified per the notification procedures in the Incident Response Action Plan and the Publication and Communication Policy.

8. Archive Storage and Retention

8.1 Archive Retention Period

Encrypted backups of terminated CA keys are retained until the last certificate issued under that key expires, plus seven (7) years. This ensures that the key material is available for forensic or legal purposes throughout the validity period of all certificates it signed, plus a substantial post-expiry buffer.

Example: If the last certificate issued under the terminated key expires on 1 January 2030, the encrypted backup is retained until 1 January 2037.

8.2 Archive Integrity Verification

The integrity of archived key backups is verified periodically:

- A SHA-256 hash of each encrypted backup file is computed at the time of archival and recorded in the Key Lifecycle Log.
- Every twelve (12) months, the Technical Lead retrieves each archived backup and verifies its SHA-256 hash against the recorded value.
- If any hash mismatch is detected, the backup is treated as potentially compromised. The incident is escalated per the Incident Response Action Plan.
- The verification event is recorded in the Key Lifecycle Log.

8.3 Archive Access

- Access to archived key backups is restricted to the Technical Lead and the Incident Commander. Both must be present to retrieve an archived key (dual access control).
- Every access to the archive storage locations is logged (cloud storage access logs for the primary location; physical access log for the secondary location).
- The passphrase required to decrypt the archived key is stored separately from the encrypted backup (physical safe at a different location). Retrieval of the passphrase also requires dual access.

8.4 Archive Disposal

Upon expiry of the retention period:

43. **Dual authorisation:** The Technical Lead and the Incident Commander both authorise the disposal in writing.
44. **Secure deletion:** The encrypted backup files are securely deleted from both storage locations. For cloud storage, this includes verifying that versioned copies are also purged. For physical media, the media is physically destroyed.
45. **Passphrase destruction:** The passphrase envelope is retrieved from the safe and destroyed (shredded) in the presence of both authorised parties.
46. **Audit log:** An audit trail entry is created (action: ca.key.archive.disposed) and the Key Lifecycle Log is updated.

9. Subscriber Key Termination

Subscriber private keys generated by the cert-management system follow a simplified termination procedure:

9.1 Conditions for Subscriber Key Termination

- Certificate revocation (for any reason).
- Certificate expiry.
- Subscriber account closure.
- Subscriber request.

9.2 Procedure

47. The certificate is revoked or confirmed expired in the cert-management database.
48. An updated CRL is published within one (1) hour of revocation.
49. The subscriber key file (subscriber-{serial}.enc) is destroyed using DestroyKey().
50. No backup of the subscriber key is retained after termination (subscriber keys are not archived).
51. The audit trail records the event (action: subscriber.key.destroyed).

Note: Subscriber private keys are not archived because they are not E-CSP signature keys. The E-CSP's obligation to archive and securely store backups applies to the CA signature private key, not to individual subscriber keys.

10. Key Lifecycle Log

The Key Lifecycle Log is a controlled record of all key lifecycle events. It supplements the system audit trail with physical witness records and archival metadata that cannot be captured by the software audit log alone.

10.1 Log Format

Field	Description
Event Date and Time	The date and time of the event in UTC.
Event Type	Generation / Backup / Termination / Archive Transfer / Archive Verification / Archive Disposal.
Key Label	The label identifying the key (e.g., 'issuing-ca').
Key Algorithm and Size	E.g., RSA 4096-bit.
Key Store Mode	Software or CloudHSM.
Authorised By	Names and roles of the authorising parties (dual authorisation).
Executed By	Name and role of the person who performed the action.
Witnesses	Names and roles of any additional witnesses present.
Backup Locations	For backup/archive events: the storage locations where the encrypted backup was placed.
Passphrase Custody	For backup events: where the passphrase envelope is stored.
SHA-256 Hash	For backup/archive events: the SHA-256 hash of the encrypted backup file.
System Audit Trail ID	The corresponding audit trail entry ID from the cert-management database.
Notes	Any additional observations or details.
Signatures	Physical signatures of the authorising parties and executor.

10.2 Log Retention

The Key Lifecycle Log is retained for the same period as the archived key backup it relates to: until the last certificate issued under the key expires, plus seven (7) years. The log is available to the auditor and the Communications Authority of Kenya upon request.

11. Roles and Responsibilities

Role	Key Lifecycle Responsibilities
Technical Lead	Executes key generation, backup, and destruction commands. Verifies backup integrity. Maintains the Key Lifecycle Log. Co-authorises all key lifecycle events (dual authorisation). Initiates archive integrity checks.
Incident Commander (Management)	Co-authorises all key lifecycle events (dual authorisation). Holds physical access to the secondary archive location and the passphrase safe. Witnesses key destruction. Approves archive disposal.
CA Operations Lead	Executes certificate issuance and revocation using the CA key during operational use. Generates CRLs, including the final CRL before key termination. Does not have independent authority to generate, backup, or destroy the CA key.
Systems/Audit Lead	Ensures the system audit trail is operational and tamper-protected. Preserves audit trail records for the retention period. Provides audit trail exports to the auditor upon request.

12. CA Key Termination Checklist

The following checklist is completed by the Technical Lead during each CA key termination event. A signed copy is filed in the Key Lifecycle Log.

#	Step	Status	Initials / Date
1	Dual authorisation obtained (Technical Lead + Incident Commander)	[ ] Complete	
2	Certificate inventory exported and reviewed	[ ] Complete	
3	Replacement CA key generated and activated (if planned termination)	[ ] Complete [ ] N/A	
4	Final CRL generated and signed	[ ] Complete	
5	Final CRL published to distribution point(s)	[ ] Complete	
6	Final CRL verified accessible and parseable	[ ] Complete	
7	Passphrase generated (min 32 characters, CSPRNG)	[ ] Complete	
8	Encrypted backup created via BackupKey() / HSM backup	[ ] Complete	
9	Backup copy 1 stored at primary location (cloud)	[ ] Complete	
10	Backup copy 2 stored at secondary location (physical off-site)	[ ] Complete	
11	Test decryption performed on backup copies	[ ] Complete	
12	SHA-256 hash of backup files recorded	[ ] Complete	
13	Passphrase sealed in tamper-evident envelope, signed by both parties	[ ] Complete	
14	Passphrase envelope stored in physical safe	[ ] Complete	
15	CA certificate PEM copied to archive locations	[ ] Complete	
16	Operational key destroyed via DestroyKey()	[ ] Complete	
17	Key file deletion verified by Technical Lead	[ ] Complete	
18	Key file deletion confirmed by Incident Commander (witness)	[ ] Complete	
19	Audit trail entries created for backup and destruction	[ ] Complete	

20	Key Lifecycle Log updated and signed	[ ] Complete	
21	Regulatory notifications sent (CA of Kenya, KE-RCA)	[ ] Complete [ ] N/A	
22	Subscriber notifications sent	[ ] Complete [ ] N/A	

13. Cross-References

Document	Relevance
Certificate Policy (CP), Section 6.2	Defines private key protection and cryptographic module engineering controls.
CPS Section 5.7	Mandates compromise and disaster recovery procedures, including key backup.
Incident Response Action Plan (CENT-ECSP-IRP-001)	Section 5.1 defines the CA private key compromise procedure, which triggers key termination.
Access Control Matrix (CENT-ECSP-ACM-001)	Defines which roles have access to the key store, key directory, and key management commands.
Publication and Communication Policy	Defines the channels for notifying Subscribers and Relying Parties of key termination events.
Audit Logging Policy	Defines the audit trail infrastructure that records key lifecycle events.
Business Continuity Plan	Addresses continuity of E-CSP operations during and after key termination, including transition to a replacement key.