White Paper

NIST 800-57 Recommendations for Key Management Requirements Analysis

cpl.thalesgroup.com

THALES
Building a future we can all trust

The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-57, Recommendations for Key Management Part 1 (Rev 5) provides guidance for cryptographic key management for U.S. Federal Government agencies. Part 1 of the publication outlines best practices for the management of cryptographic keys and discusses key management issues that must be addressed with using cryptography.

Importance of Securing Cryptographic Keys

The security of cryptographic processes is dependent on the security of the cryptographic keys used to encrypt the data. If the keys used to encrypt data are stolen with the encrypted data, the data is not secure because it can be deciphered and read in plain text.

NIST emphasizes the importance of protecting cryptographic keys in the publication, "The proper management of cryptographic keys is essential to the effective use of cryptography for security. Poor key management may easily compromise strong algorithms."[1] NIST states that:

> *Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of cryptographic mechanisms and protocols associated with the keys, and the protection provided to the keys. Secret and private keys need to be protected against unauthorized disclosure, and all keys need to be protected against modification.* [1]

For encryption to successfully secure sensitive data, the cryptographic keys themselves must be secured, managed and controlled by your organization and not a third-party or cloud provider. As agencies deploy ever-increasing numbers of siloed encryption solutions, they find themselves managing inconsistent policies, different levels of protection, and escalating costs.

The simplest path through this maze is to transition to a centralized key management model. Encryption key management involves administering the full lifecycle of cryptographic keys and protecting them from loss or misuse. Keys have a life cycle: They're created, live useful lives, and are retired. Key lifecycle management includes generating, using, storing, distributing, archiving, and deleting keys.

CipherTrust Data Security Platform

CipherTrust Data Security Platform (CDSP) offers a unified, scalable solution for protecting sensitive data across on-premises, cloud, and hybrid environments. Designed to simplify and strengthen data security, the platform centralizes critical capabilities such as data discovery, encryption, access controls, and compliance reporting.

CDSP is managed by CipherTrust Manager (CM), which is a centralized key management solution for enterprises, providing a unified console for managing encryption keys across on-premises and cloud environments. It simplifies key lifecycle management, including generation, backup, and deletion, while enforcing role-based access control and security policies. It also offers features like multi-tenancy support, robust auditing, and a REST API for custom solutions.

A cornerstone of CDSP is its robust key management functionality, which empowers organizations to securely create, store, rotate, and manage encryption keys across diverse systems and services. With support for FIPS 140-2 L1-compliant hardware security modules (HSMs) and integrations with major cloud providers, CDSP ensures that encryption keys remain under enterprise control—reducing risk and helping meet regulatory mandates.

CDSP streamlines operations with automated policies and centralized oversight, allowing security teams to confidently enforce data protection without disrupting performance. Whether securing databases, files, containers, or applications, CDSP delivers a comprehensive, policy-driven framework for managing and safeguarding critical data and the cryptographic keys that protect it.

Platform Benefits

Robust Key Management, Encryption and Tokenization

CDSP offers encryption solutions to protect data while it is stored on-premises, in the cloud, or in backups. CDSP supports transparent encryption, which means that data can be encrypted without requiring changes to applications or workflows. CDSP provides centralized key

[1] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

management to ensure that encryption keys are securely generated, stored, and managed. Tokenization replaces sensitive data with non-sensitive tokens, making it difficult for unauthorized individuals to access or misuse the data. CDSP supports dynamic data masking which allows sensitive data to be masked or redacted in real time, preventing unauthorized access even when data is in use.

Take Control of Your Sensitive Data Across Clouds

CDSP provides solutions for protecting data in public, private, and hybrid cloud environments. The platform integrates with popular cloud platforms such as AWS, Azure, and GCP. Centralizes multi cloud key management for BYOK, HYOK and cloud native encryption keys across any combination of clouds and on-premises with a single UI.

Integrate Security into Development improving DevSecOps efficiency

CDSP can be integrated into DevOps workflows to ensure that security is built into applications from the beginning. The platform provides solutions for protecting sensitive data during development, testing, and deployment. It rapidly secures, deploys and runs cloud-native workloads across cloud service providers and speed-up continuous integration and continuous delivery processes.

Managing Compliance and Risk

As part of CDSP, Data Risk Intelligence provides visibility into where your data sources that contain sensitive data are. It identifies the encryption status of each data source and offers recommendations on how to meet encryption goals. CDSP helps organizations comply with data protection regulations such as GDPR, HIPAA, and PCI DSS. By protecting sensitive data, CDSP can help organizations reduce the risk of data breaches and other security incidents. File Activity Monitoring (FAM) delivers real-time visibility into file usage, encryption status, and sensitive data access, enabling proactive threat detection and data protection across your organization.

Manage and Automate Access to Secrets

CipherTrust Secrets Management (CSM) offers a range of capabilities to securely store, manage, and distribute secrets across different developer environments, ensuring that they are protected from unauthorized access and misuse. It improves developer efficiency and centralizes management for all secret types with scalable SaaS.

Discovering and Classifying your Sensitive Data

CDSP helps organizations discover and classify sensitive data across their entire infrastructure, including files, databases, and big data. It provides visibility into where sensitive data resides and how it is being accessed, helping organizations assess their risk and prioritize protection efforts. It automatically scans and detects code containing sensitive information (API keys, tokens, passwords, etc.) mistakenly included by developers.

NIST SP 800-57 Requirements Mapping

Focusing on the capabilities needed to meet the requirements outlined in NIST SP 800-57, the following table provides details on CipherTrust Platform.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6	**Protection Requirements for Key Information**		
6.1	Protection and Assurance Requirements	√	CipherTrust Manager is available in both virtual and physical appliances, offering FIPS 140-2 L2 certifications to meet diverse security requirements. Virtual deployments can integrate with various network-accessible Hardware Security Modules (HSMs) to establish a FIPS 140 Level 3 root of trust. CipherTrust Manager supports deployment on-premises and across private or public cloud environments. For enhanced protection, its disk can be fully encrypted, with encryption initiated either during initial launch or applied to an already running instance.
6.1.1	Summary of Protection and Assurance Requirements for Cryptographic Keys	√	
6.1.2	Summary of Protection Requirements for Other Related Information	√	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6.2	Protection Mechanisms	✓	CipherTrust Manager is available as both virtual and physical appliances, offering FIPS 140-2 L2 certifications. The appliance supports full disk encryption and can be deployed in high-availability, Active/Active clustered configurations. Real-time replication of keys, policies, and configurations enables seamless disaster recovery and business continuity across multiple instances.
6.2.1	Protection Mechanisms for Key Information in Transit	✓	
6.2.1.1	Availability	✓	CipherTrust Manager ensures high availability through clustered deployment options and redundant communication paths between nodes.
6.2.1.2	Integrity	✓	Supports secure transmission of key materials and policies using TLS/SSL to maintain integrity during transit.
6.2.1.3	Confidentiality	✓	Utilizes TLS-encrypted communications and optional client authentication to ensure confidentiality of key exchanges and administrative sessions.
6.2.1.4	Association with Usage or Application	✓	Supports PKCS #11, JCE, .NET, and the Key Management Interoperability Protocol (KMIP) to bind keys to specific usage contexts and applications. Additionally, the SafeNet REST API allows integration with custom applications using enterprise-grade key management
6.2.1.5	Association with Other Entities	✓	Keys can be explicitly associated with users, applications, or services through role-based and attribute-based access controls.
6.2.1.6	Association with Other Related Key Information	✓	Offers granular access control via Attribute-Based Access Control (ABAC), ensuring that keys are only used in authorized contexts. Also supports: • Secure key distribution using SSL • Secure storage of key encryption keys in Luna Network HSMs
6.2.2	Protection Mechanisms for Key Information in Storage	✓	
6.2.2.1	Availability	✓	High-availability clustering and replication ensure keys remain accessible across failovers and maintenance events.
6.2.2.2	Integrity	✓	Key material is stored securely and verified using checksums and cryptographic validation mechanisms.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6.2.2.3	Confidentiality	✓	When integrated with CipherTrust Transparent Encryption (CTE), keys are securely passed over encrypted channels. Admin access occurs via HTTPS, and communication between agents and the manager is encrypted using ephemeral session keys, minimizing exposure and enhancing confidentiality.
6.2.2.4	Association with Usage or Application	✓	CipherTrust Manager binds keys to specific applications or data sets through policy-driven associations and APIs, ensuring precise usage control.
6.2.2.5	Association with Other Entities	✓	Keys are mapped to users, roles, or services through defined policies and fine-grained access controls.
6.2.2.6	Association with Other Related Key Information	✓	Relationships between keys and associated metadata (e.g., rotation dates, ownership, tags) are preserved and managed via policy
6.2.3	Metadata for Keys	✓	Keys managed by CipherTrust Manager contain a single metadata record, which can link to multiple versions of the key. Each version is uniquely tracked, providing a detailed audit trail for compliance and lifecycle management.
7	**Key States and Transitions**		
7.1	Pre-activation State	✓	CipherTrust Manager provides centralized key lifecycle management for all CipherTrust Data Security Platform products. Built on an extensible microservices architecture, it supports all standard key states—from generation through retirement—ensuring secure and auditable transitions throughout the lifecycle. Key management capabilities include: • State transitions such as activation, suspension, deactivation, compromise, and destruction • Role-based access control (RBAC) for fine-grained key and policy management • Multi-tenancy support to isolate keys and operations across users or departments • Comprehensive auditing and reporting for key usage, operational changes, and policy enforcement CipherTrust Manager also supports secure key generation, backup and recovery, and policy-driven actions to manage compromised or decommissioned keys.
7.2	Active State	✓	
7.3	Suspended State	✓	
7.4	Deactivated State	✓	
7.5	Compromised State	✓	
7.6	Destroyed State	✓	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8	**Key-Management Phases and Functions**		
8.1	Pre-operational Phase	√	The CipherTrust Data Security Platform supports comprehensive key management functions that are logically separated from standard operations. These functions include domain creation, key creation, host registration, system initialization, and audit logging.
8.1.1	Entity Registration Function	√	In multi-domain environments, users and administrators are logically isolated. Admins can select different encryption algorithms and key lengths per domain, increasing flexibility and control. Supported algorithms include AES-128 and AES-256.
8.1.2	System Initialization Function	√	
8.1.3	Initialization Function	√	
8.1.4	Keying-Material Installation Function	√	Encrypted communications between CipherTrust Manager (CM) and agents are configurable. The platform supports TLS 1.2+ using its REST API to ensure secure key delivery and API access. Key establishment is handled through CipherTrust Transparent Encryption (CTE) or agent-based solutions, where selectable encryption methods are applied per deployment requirement.
8.1.5	Key Establishment Function	√	
8.1.5.1	Generation and Distribution of Asymmetric Key Pairs	√	Secure control paths exist between: • The daemon on the host • The SecFS kernel module responsible for system-level security Steps for key pair generation: 1. CM creates a public/private key pair 2. A Certificate Signing Request (CSR) is generated 3. The signed certificate is returned and stored in CM's secure database When operating in agent mode, the user space portion of the agent creates a local key pair and sends a CSR to CM. The signed certificate is returned and installed to complete registration. The kernel space portion also generates a key pair, and sends its public key to CM to complete secure host provisioning.
8.1.5.1.1	Distribution of Public Keys	√	
8.1.5.1.1.1	Distribution of a Trust Anchor's Public Key in a PKI	√	
8.1.5.1.1.2	Submission to a Registration Authority or Certification Authority	√	
8.1.5.1.1.3	General Distribution of Static Public Keys	√	
8.1.5.1.2	Distribution of Ephemeral Public Keys	√	
8.1.5.1.3	Distribution of Centrally Generated Key Pairs	√	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.1.5.2	Generation and Distribution of Symmetric Keys	✓	Key generation supports standards-compliant symmetric algorithms and configurable key lengths Key distribution is supported via: Manual distribution (for controlled environments) Automated distribution and key wrapping Keys are associated with policies that define lifecycle states, access controls, and allowed usage
8.1.5.2.1	Key Generation	✓	
8.1.5.2.2	Key Distribution	✓	
8.1.5.2.2.1	Manual Key Distribution	✓	
8.1.5.2.2.2	Automated Key Distribution/Key Transport/Key Wrapping	✓	
8.1.5.2.2.3	Key Agreement	✓	Support for key agreement protocols can be configured via CipherTrust Manager's API and agent-based workflows, depending on the environment and policy enforcement needs.
8.1.5.3	Generation and Distribution of Other Keying Material	✓	CipherTrust Manager supports the secure generation and distribution of other cryptographic materials, such as initialization vectors, shared secrets, RBG seeds, random numbers, and passwords. These are passed between the CipherTrust Manager and hosts via a one-time-use AES-256 random key generated by the Manager. The random key encrypts the desired encryption keys. This AES key is then encrypted using the kernel-space public key of the host. The encrypted payload is delivered to the host, where the kernel-space private key decrypts the random key. The decrypted random key is used to decrypt the actual encryption keys, which are then applied to the target file, directory, or executable. This method ensures secure in-transit handling of sensitive materials like initialization vectors, shared
8.1.5.3.1	Domain Parameters	✓	
8.1.5.3.2	Initialization Vectors	✓	
8.1.5.3.3	Shared Secrets	✓	
8.1.5.3.4	RBG Seeds	✓	
8.1.5.3.5	Other Public and Secret Information	✓	
8.1.5.3.6	Intermediate Results	✓	
8.1.5.3.7	Random Bits/Numbers	✓	
8.1.5.3.8	Passwords	✓	
8.1.6	Key Registration Functiony	✓	CipherTrust Manager supports automated key registration and lifecycle management, allowing keys to be registered during generation or securely imported from external systems. Registered keys are tagged, versioned, and governed by policies such as expiration, usage, and rotation.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2	Operational Phase	✓	CipherTrust Manager provides comprehensive tools for managing cryptographic keys throughout their operational lifecycle. This includes: • Key generation, import, export, and rotation • Support for symmetric and asymmetric key types • Control over key size, usage constraints, and policy enforcement All keys are stored in a centralized, hardware or virtual appliance, simplifying operations while enforcing strong access controls and auditability. CipherTrust Manageralso supports Active/Active clustering for high availability, ensuring continuous uptime for encryption and key management services.
8.2.1	Normal Operational Storage Function	✓	
8.2.1.1	Cryptographic Module Storage	✓	All keys are securely stored inside CipherTrust Manager, a FIPS-certified cryptographic module, ensuring isolation, protection, and compliance with international standards.
8.2.1.2	Immediately Accessible Storage Media	✓	Keys are readily available to authorized services and applications during runtime, while access is tightly controlled through policy-based authorization, role-based access control (RBAC), and secure API communication.
8.2.2	Continuity of Operations Function	✓	CipherTrust Manager supports high-availability deployments, including Active/Active configurations, for continuous access to keys and services. This architecture allows for automatic failover and synchronized replication, ensuring that encryption and key management operations continue uninterrupted even during system events or outages.
8.2.2.1	Backup Storage	✓	CipherTrust Manager uses backup encryption keys to securely protect key backups. Administrators can: • Generate a new backup encryption key, or • Reuse an existing key when creating backups Backup keys are stored encrypted using a secure vault within CipherTrust Manager. Each backup may include critical data such as user keys or database dumps, all encrypted according to the system's configured key hierarchy. A final encrypted backup file is generated using the designated backup key, ensuring secure recovery and compliance with retention and protection policies.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2.2.2	Key Recovery Function	√	CipherTrust Manager supports robust key recovery capabilities, including key generation, backup, restore, deactivation, and deletion. Keys are versioned and archived, supporting rapid recovery when needed. With version control and Thales Live Data Transformation (CTE-LDT), restored keys can be automatically aligned to historical data states. During recovery, keys retrieved from backup are securely applied to the relevant datasets, with encryption maintained using current cryptographic policies.
8.2.3	Key Change Function	√	CipherTrust Manager supports: Key change for replacing compromised or outdated keys Re-keying, enabling seamless replacement of keys without interrupting operations Key update workflows that apply newer keys to older datasets as part of a versioned lifecycle process All updates are governed by policy-based controls and audit logging for traceability.
8.2.3.1	Re-keying	√	
8.2.3.2	Key Update Function	√	
8.2.4	Key Derivation Methods	√	Enforces secure key derivation methods, centralizes control, automates policy enforcement, and ensures compliant key lifecycle management.
8.3	Post-Operational Phase	√	
8.3.1	Key Archive and Key Recovery Functions	√	Securely archiving keys, enabling authorized recovery, enforcing access controls, and ensuring compliance with cryptographic key management policies.
8.2.3	Key Change Function	√	CipherTrust Manager supports: • Key change for replacing compromised or outdated keys • Re-keying, enabling seamless replacement of keys without interrupting operations • Key update workflows that apply newer keys to older datasets as part of a versioned lifecycle process All updates are governed by policy-based controls and audit logging for traceability.
8.2.3.1	Re-keying	√	
8.2.3.2	Key Update Function	√	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2.4	Key Derivation Methods	✓	CipherTrust supports secure key derivation methods consistent with industry standards. Derived keys are created based on unique, application-specific inputs and comply with approved cryptographic standards for secure usage.
8.3	Post-Operational Phase	✓	CipherTrust Manager retains cryptographic materials per policy after their active use has ended, allowing for secure archival, auditing, or decommissioning.
8.3.1	Key Archive and Key Recovery Functions	✓	CipherTrust Manager supports secure archiving of retired or inactive keys. Archived keys remain accessible for approved recovery scenarios and are stored using encryption consistent with active key policies. Archived keys can be restored based on versioned metadata Policy controls govern access and retent
8.3.2	Entity De-registration Function	✓	CipherTrust Manager allows for full lifecycle control of clients (e.g., ProtectFile, ProtectV, CTE). Administrators can: • Register and manage client entities (excluding ProtectFile clients) • View, modify, revoke, or delete client registrations as needed Once a client is deregistered, all communication with CipherTrust Manager is terminated, ensuring immediate and complete disassociation.
8.3.3	Key De-registration Function	✓	Within CipherTrust Manager there is a Key Admins group. Key Administrators have permissions to managing keys on the system. They can: • create or modify their own keys • perform key management operations on keys created by all users on the system There is a System Defined Group named "CTE Admins". Users within the "CTE Admins" group are CTE Administrators. Only users of the "CTE Admins" group can delete CTE keys. Depending on the Key Management usecase a specified admin within CipherTrust Manager will be able be to manage keys or delete keys that are not in use.
8.3.4	Key Destruction Function	✓	
8.3.5	Key Revocation Function	✓	
8.4	Destroyed Phase	✓	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.3.3	Key De-registration Function	✓	Within CipherTrust Manager, a designated Key Admins group manages key de-registration. Admins have permissions to: • Create and manage their own keys • Perform operations on other keys based on their assigned privileges Key de-registration removes keys from operational use but retains metadata for audit if required.
8.3.4	Key Destruction Function	✓	Only users in the System Defined Group: "CTE Admins" have permission to permanently destroy keys (such as CTE policies and associated keys). Keys marked for deletion are securely removed based on policy and usage state Only unused or expired keys can be deleted, ensuring cryptographic continuity
8.3.5	Key Revocation Function	✓	CipherTrust Manager supports key revocation based on usage status or compromise events. Revoked keys are: • Immediately marked as inactive • Prevented from being used in future encryption or decryption operations • Tracked in audit logs for compliance and incident response
8.4	Destroyed Phase	✓	The Destroyed Phase ensures that cryptographic keys are irretrievably deleted when no longer needed. Only authorized administrators (e.g., CTE Admins) can initiate destruction Keys must be inactive or non-operational before destruction Destruction operations are logged to maintain audit integrity
9	**Additional Considerations**		
9.1	Access Control and Identity Authentication	✓	CipherTrust Manager enhances access control through kernel-level agents and AES-256 encryption, surpassing typical OS-level controls. It enforces least privilege access, allowing only authenticated administrators and clients to communicate with the Manager. Integration with role-based access controls ensures secure, policy-driven identity management.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.2	Inventory Management	✓	CipherTrust Manager provides centralized inventory visibility and lifecycle management for all cryptographic keys and certificates across the CipherTrust Data Security Platform. Its microservices-based architecture simplifies operations such as key generation, rotation, backup, archival, and deletion.
9.2.1	Key Inventories	✓	All cryptographic keys are cataloged with metadata such as key type, usage, status, creation date, and expiration, enabling full visibility and control across environments.
9.2.2	Certificate Inventories	✓	CipherTrust Manager tracks certificates used for encryption, authentication, and digital signing. Each certificate is indexed for quick retrieval and lifecycle management.
9.3	Accountability	✓	All user and administrative actions within CipherTrust Manager are fully auditable and attributable through detailed logs. These logs include timestamps, user roles, affected objects (e.g., keys, policies), and the nature of each action, helping ensure accountability and support for internal or external audits.
9.4	Audit	✓	CipherTrust Manager generates comprehensive audit logs capturing all key lifecycle operations (e.g., generation, access, modification, rotation, deletion). Logs are tamper-resistant and can be forwarded to external SIEM systems or retained locally for compliance. This capability ensures full traceability for audit and compliance purposes.
9.5	Key-Management System Survivability	✓	CipherTrust Manager supports high availability through clustered deployments with real-time replication of keys, policies, and configuration data across appliances. This architecture supports: • Seamless failover and disaster recovery • No disruption to cryptographic operations • Centralized key management across distributed environments with minimal performance impact

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.5.1	Backed Up and Archived Key	✓	CipherTrust Manager automatically creates versioned backups of all cryptographic keys. Backups: • Are stored securely on the appliance • Remain available until explicitly deleted • Can be restored on the same or another appliance • Are always encrypted using a designated backup key If a specific backup key is not provided, a default backup key is used to ensure data protection and future recovery.
9.5.2	Key Recovery	✓	During key recovery, backup files—encrypted using the assigned or default backup key—can be: • Downloaded from the appliance • Re-uploaded to the same or a different CipherTrust Manager appliance • Decrypted and restored using the required backup key This enables secure restoration of archived encryption keys for operational continuity and compliance with retention policies.
9.5.3	System Redundancy/Contingency Planning	✓	CipherTrust Manager supports robust system redundancy and contingency planning through clustered deployments with real-time replication of keys, policies, and configurations across multiple appliances. This setup ensures high availability, seamless failover, and disaster recovery Organizations with multiple encryption endpoints can centrally manage keys without disrupting system performance Business continuity is preserved even during infrastructure failure scenarios

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.5.3.1	General Principles	√	CipherTrust Manager supports Active/Active high availability deployments to ensure minimal downtime. Designed for 24x7 operations with full support for uninterrupted key management and encryption services All keys and related policies are versioned and securely stored, allowing seamless recovery in the event of failure or compromise
9.5.3.2	Cryptography and Key-Management-Specific Recovery Issues	√	CipherTrust Manager maintains comprehensive backups and versioning of keys. These capabilities allow: • Recovery of archived encryption keys aligned with the original version of protected data • Secure application of keys to historical datasets through Live Data Transformation, preserving data integrity • Controlled and auditable recovery procedures in line with cryptographic best practices
9.5.4	Compromise Recovery	√	In the event of a key compromise, CipherTrust Manager enables recovery via Live Data Transformation, which allows data encrypted with a compromised key to be seamlessly re-encrypted using a new, secure key. This eliminates the need to re-ingest or decrypt raw data manually Recovery operations are automated, secure, and aligned with enterprise-grade data protection policies

About Thales

Thales is a global leader in cybersecurity, helping the most trusted organizations protect critical applications, data, identities, and software anywhere, at scale. Through Thales' integrated platforms, customers achieve better visibility of risks, defend against cyber threats, close compliance gaps, and deliver trusted digital experiences for billions of consumers every day.

THALES

Building a future we can all trust

Contact us

For contact information,
please visit cpl.thalesgroup.com/contact-us

cpl.thalesgroup.com