Shield Platform Encryption

Salesforce, Spring '25

Last updated: January 10, 2025

CONTENTS

STRENGTHEN YOUR DATA'S SECURITY WITH SHIELD PLATFORM ENCRYPTION

Shield Platform Encryption gives your data a whole new layer of security while preserving critical platform functionality. You can encrypt sensitive data at rest, not just when transmitted over a network, so your company can confidently comply with privacy policies, regulatory requirements, and contractual obligations for handling private data.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Shield Platform Encryption builds on the classic encryption options that Salesforce offers all license holders. Data stored in many standard and custom fields and in files and attachments is encrypted using an advanced hardware security module (HSM)-based key derivation system. So it's protected even when other lines of defense are compromised.

Your data encryption key material is never saved or shared across orgs. You can choose to have Salesforce generate key material for you, or you can upload your own. By default, Shield Platform Encryption uses a key derivation function (KDF) to derive data encryption keys on demand from a primary secret and your org-specific key material. It then stores that derived data encryption key (DEK) in an encrypted key cache. DEKs are never stored on disk, and your org-specific key material is always wrapped.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

You can also opt out of key derivation on a key-by-key basis. Or you can store your DEK outside of Salesforce and have either the External Key Management service or the Cache-Only Key Service fetch it on demand from a key service that you control. The DEKs that you provide are always wrapped. No matter how you choose to manage your keys, Shield Platform Encryption secures your key material at every stage of the encryption process.

You can try out Shield Platform Encryption at no charge in Developer Edition orgs. It's available in sandboxes after it's provisioned for your production org.

Tip: Whether you're using Shield Platform Encryption or Classic Encryption, you can track the encryption policy status across your entire org. It's a simple process with the Security Center app, which can capture many useful security metrics. See Take Charge of Your Security Goals with Security Center.

What You Can Encrypt

Shield Platform Encryption lets you encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, Salesforce search indexes, and more. We continue to make more fields and files available for encryption.

Platform Encryption Q&A

Here are some frequently asked questions about platform encryption.

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a primary secret that Salesforce maintains. By default, we combine these secrets to create your unique data encryption key (DEK). You can also supply your own final DEK. We use your DEK to encrypt data that your users put into Salesforce, and we use it to decrypt data when your authorized users need it.

Set Up Your Encryption Policy

An encryption policy is your plan for encrypting data with Shield Platform Encryption. You can choose how you want to implement it. For example, you can encrypt individual fields and apply different encryption schemes to those fields. Or you can choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption policy.

Filter Encrypted Data with Deterministic Encryption

You can filter data that's protected with Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. You can apply case-sensitive deterministic encryption or exact-match case-insensitive deterministic encryption to data on a field-by-field basis.

Key Management and Rotation

With Shield Platform Encryption, you control and rotate the key material used to encrypt your data. You can use Salesforce to generate a tenant secret for you, which is then combined with a primary secret for each release to derive a data encryption key. This derived data encryption key is then used in encryption and decryption functions. You can also use the Bring Your Own Key (BYOK) service to upload your own key material. Or you can store your key material outside of Salesforce. Use the External Key Management Service or the Cache-Only Key Service to fetch your key material on demand.

Shield Platform Encryption Customizations

Some features and settings require adjustment before they work with encrypted data.

Tradeoffs and Limitations of Shield Platform Encryption

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

What You Can Encrypt

Shield Platform Encryption lets you encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, Salesforce search indexes, and more. We continue to make more fields and files available for encryption.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Which Standard Fields Can I Encrypt?

You can encrypt certain fields on standard and custom objects, data in Chatter, and search index files. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

Which Custom Fields Can I Encrypt?

You can apply Shield Platform Encryption to the contents of fields that belong to one of these custom field types.

Which Files Are Encrypted?

When you enable Shield Platform Encryption for files and attachments, all files and attachments that can be encrypted are encrypted. The body of each file or attachment is encrypted when it's uploaded.

What Other Data Elements Can I Encrypt?

In addition to standard and custom field data and files, Shield Platform Encryption supports other Salesforce data. You can encrypt CRM Analytics datasets, Chatter fields, fields in the Salesforce B2B Commerce managed package, and more.

Which Standard Fields Can I Encrypt?

You can encrypt certain fields on standard and custom objects, data in Chatter, and search index files. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

When you encrypt a field, existing values aren't encrypted immediately. Values are encrypted only after they're touched or after they're synchronized with the latest encryption policy. Synchronize existing data with your policy from Setup on the Encryption Statistics page.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

Compatible Standard Fields

You can encrypt the contents of these standard field types.

Object	Fields	Notes
Account Participant	Comments	The Account Participant object is available in select Salesforce Industries products.
Accounts	Account Name Account Site Billing Address (encrypts Billing Street and Billing City) Description Fax Phone Shipping Address (encrypts Shipping Street and Shipping City) Website	If you enabled Person Accounts, certain account and contact fields are combined into one record. In that case, you can enable encryption for a different set of Account fields.
Accounts with Person Accounts enabled	Account Name Account Site Assistant Assistant Phone Billing Address (encrypts Billing Street and Billing City) Description Email Fax Home Phone	

Object	Fields	Notes
	Mailing Address (encrypts Mailing Street and Mailing City) Mobile Other Address (encrypts Other Street and Other City) Other Phone Phone Shipping Address (encrypts Shipping Street and Shipping City) Title Website	
Activity	Description (encrypts Event—Description and Task—Comment) Subject (encrypts Event—Subject and Task—Subject)	Selecting an Activity field encrypts that field on standalone events, event series (Lightning Experience), and recurring events (Salesforce Classic).
AI Natural Language Process Chunk Result	Additional Information Response	
AI Natural Language Process Result	Additional Information Response	
Applicant	Birth Date Email First Name Last Name Middle Name Phone Prefix Suffix Business Entity Name Unique Reference Number	
Application Form	Submission Date	
Application Form Participant	Comment	
Application Form Product Participant	Comment	

Object	Fields	Notes
Assessment Question Response	Choice Value Date Value Date Time Value Response Text Response Value	
Authorization Form	Name	
Authorization Form Consent	Name	
Authorization Form Data Use	Name	
Authorization Form Text	Name	
Business License	Identifier	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Business License Application	Site Address (encrypts Site Street and Site City)	
Business Profile	Business Operating Name Business Tax Identifier	
Cases	Description Subject	
Case Comments	Body (including internal comments)	
Chat Transcript	Body Supervisor Transcript Body	Before you can apply encryption to Chat fields, add the Supervisor Transcript Body field to the LiveChatTranscript record home layout.
Contact Point Address	Address	
Contact Point Email	Email address	
Contact Point Phone	Telephone number	
Contacts	Assistant Assistant Phone Description Email Fax Home Phone Mailing Address (encrypts Mailing Street and Mailing City)	

Object	Fields	Notes
	Mobile Name (encrypts First Name, Middle Name, and Last Name) Other Address (encrypts Other Street and Other City) Other Phone Phone Title	
Contracts	Billing Address (encrypts Billing Street and Billing City) Shipping Address (encrypts Shipping Street and Shipping City)	
Conversation Context Entry	Key Value	
Conversation Entry	Message	
Conversation Participant	Participant Display Name	
Course Offering	Name	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Custom Objects	Name	
Email Messages	From Name From Name To Address CC Address BCC Address Subject Text Body HTML Body Headers	If you use Email-to-Case, these fields are also encrypted on the customer emails that generate cases.
Email Message Relations	Relation Address	
Flow Orchestration Work Item	Screen Flow Inputs	
Identity Document	Document Number	

Object	Fields	Notes
	Expiration Date Issue Date	
Individual	Name	The Individual object is available only if you enable the setting to make data protection details available in records.
Leads	Address (Encrypts Street and City) Company Description Email Fax Mobile Name (Encrypts First Name, Middle Name, and Last Name) Phone Title Website	
List Emails	From Name From Address Reply To Address	
List Email Sent Results	Email	
Loan Applicant	Loan Applicant Name	
Loan Applicant Address	Residence Address	
Messaging End User	Messaging Platform Key Name Profile Picture URL	
OCR Document Scan Result	Extracted Values	
OCR Scan Result Template Mapping	Mapped Fields	
Opportunities	Description Next Step Opportunity Name	

Object	Fields	Notes
Opportunity Participant	Comments	The Opportunity Participant object is available in select Salesforce Industries products.
Party Profile Participant	Comment	
Payment Instrument	Bank Account Name	—
Public Complaint	Business Address Business Name Email First Name Last Name Mobile Number	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Recommendations	Description	
Referral	Client Email Client Name Client Phone Provider Email Provider Name Provider Phone Referrer Email Referrer Name Referrer Phone	
Regulatory Code Violation	Corrective Action Description Description	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Service Appointments	Address (Encrypts Street and City) Description Subject	
Social Persona	Bio Profile URL Provider External Picture URL Real Name	Before you can apply encryption to Social Persona fields, make sure that Social Customer Service is enabled and connected to a Marketing Cloud Engagement social service.

Object	Fields	Notes
Social Post	Attachment URL Headline Message Post URL Social Handle	Before you can apply encryption to Social Post fields, make sure that Social Customer Service is enabled and connected to a Marketing Cloud Engagement social service.
Survey Question Response	Date Value Date Time Value Choice Value Response Value	
Training Course	Description Name	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
User	Email	
Utterance Suggestion	Utterance	
Video Call	Description End Date Time Start Date Time Vendor Meeting Uuid	
Video Call Participant	Email Join Date Time Leave Date Time	
Violation Enforcement Action	Description	Emergency Response Management for Public Sector standard objects and fields are available to users who have the Emergency Response for Public Sector permission set license.
Voice Call	FromPhoneNumber ToPhoneNumber	
Web Quote	Introduction Notes Ship to City	

Object	Fields	Notes
	Ship to Country Ship to Name Ship to Postal Code Ship to State Ship to Street Description Product Code	
Work Orders	Address (Encrypts Street and City) Description Subject	
Work Order Line Items	Address (Encrypts Street and City) Description Subject	

Compatible Automotive Cloud Fields

Automotive Cloud standard objects and fields are available to users who have the Automotive Foundation User and the Vehicle and Asset Finance permission sets.

Object	Fields
Financial Account	Financial Account Number Name

Compatible Health Cloud Fields

Health Cloud standard objects and fields are available to users who have the Health Cloud Platform permission set license.

Note: Deterministic encryption is unavailable for long text fields and fields that have Notes in the name.

Object	Fields
Care Plan Template Problem	Name
Care Program Enrollee	Name
Care Program Enrollee Product	Name

Object	Fields
Care Program Provider	Name
Care Request	Admission Notes Disposition Notes Facility Record Number First Reviewer Notes Medical Director Notes Member First Name Member Last Name Member ID Member Group Number Resolution Notes Root Cause Notes
Care Request Drug	Prescription Number
Care Specialty	Name
Contact Encounter	Name
Coverage Benefit	Benefit Notes Coinsurance Notes Copay Notes Deductible Notes Lifetime Maximum Notes Name Out-of-Pocket Notes Source System Identifier
Coverage Benefit Item	Coverage Level Notes Service Type Service Type Code Source System Identifier
Healthcare Provider Specialty	Name
Healthcare Provider Treated Condition	Name

Object	Fields
Member Plan	Affiliation Group Number Issuer Number Member Number Name Primary Care Physician Source System Identifier
Purchaser Plan	Name

Compatible Financial Services Cloud Fields

Financial Services Cloud standard objects and fields are available to users who have Financial Services Cloud enabled.

Object	Fields
Application Form Seller Item	Vehicle Identification Number Engine Number Vehicle Registration Number PropertyAddress Scheduled Delivery Date Property UnitI dentifier Make Model Trim
Application Form Vendor Product	Address
Custom Object Participant	Comments
Financial Deal	Description Financial Deal Code Name
Financial Deal Asset	Address
Financial Deal Bid	Bid Round
Financial Deal Interaction	Comment
Financial Deal Interaction Summary	Comment

Object	Fields
Interaction	Description Name
Interaction Attendee	Email Address
Interaction Summary	Meeting Notes Next Steps Name
Interaction Related Account	Comment
Interaction Summary	Next Steps Meeting Notes Title
Interaction Summary Discussed Account	Comment
Party Financial Asset Lien	Lien Holder Maturity Date
Party Financial Liability	Start Date Term Lender Liability Account Identifier
Party Profile	Name Full Name First Name Middle Name Last Name Party Identification Name Primary Identifier Business Entity Name Primary Identification Name Primary Identifier
Payment Mandate	Mandate Submission Date Mandate End Date Mandate Internal Identifier Mandate External Identifier

Object	Fields
	Mandate Effective Date Bank Account Number Bank Routing Number Disbursement Address Bank Branch Address

Compatible Grantmaking Fields

Grantmaking standard objects and fields are available to users who have Grantmaking enabled.

Object	Fields
Budget Participant	Comments
Funding Award Participant	Comments
Funding Opportunity Participant	Comments
Individual Application Participant	Comments
Individual Application Task Participant	Comments

Compatible Insurance for Financial Services Cloud Fields

Insurance for Financial Services Cloud standard objects and fields are available to users who have Financial Services Cloud enabled.

Object	Fields
Business Milestone	Milestone Description Milestone Name
Claim	Claim Number Incident Site Report Number
Customer Property	Address Lien Holder Name
Insurance Policy	Policy Number Servicing Office Universal Policy Number
Person Life Event	Event Description

Object	Fields
	Event Name
Securities Holding	Name

Compatible Loyalty Management Fields

Loyalty Management standard objects and fields are available to users who have Loyalty Management enabled.

Shield Platform Encryption Supported Objects	Fields
Loyalty Program Group Member Relationship	Member Name

Compatible Nonprofit Cloud Fields

Nonprofit Cloud standard objects and fields are available to users who have Nonprofit Cloud features enabled.

Object	Fields
Gift Entry	City Country Email Expiry Month Expiry Year First Name Home Phone Last 4 Last Name Mobile Phone Organization Name State/Province Street
Payment Instrument	Bank Account Number

Compatible Public Sector Solution Fields

Public Sector Solutions standard objects and fields are available to users who have Public Sector Solutions features enabled.

Object	Fields
Application Form Evaluation Participant	Comments

Object	Fields
Case Proceeding Participant	Comments
Complaint Participant	Comments
Recruitment Requisition Participant	Comments

Compatible Salesforce CPQ Fields

Salesforce CPQ standard objects and fields are available to users who have the Salesforce CPQ permission set license.

Object	Fields
Lookup Data	Lookup Data
Process Input Value	Value
Quote	Bill To City Bill To Country Bill To Name Bill To Postal Code Bill To State Bill To Street Introduction Notes Ship To City Ship To Country Ship To Name Ship To Postal Code Ship To State Ship To Street
Quote Template	Company Name
Quote Term	Body
Tax Exemption Certificate	Certificate Number Country County Exempt Company Name Notes Postal Code State

Object	Fields
	Street Address Street Address_2

Compatible Workplace Command Center Fields

Object	Fields	Notes
Employee	Alternate Email Email First Name Home Address Home Phone Last Name Middle Name Preferred First Name Work Phone	To enable encryption on the Employee object, contact Salesforce Customer Support.

SEE ALSO:

Set Up Field-Level Encryption

Which Custom Fields Can I Encrypt?

You can apply Shield Platform Encryption to the contents of fields that belong to one of these custom field types.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

- Email
- Phone
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich)
- URL
- Date
- Date/Time

Note: To enable encryption on any custom object, you navigate directly to the object in Object Manager

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

After a custom field is encrypted, you can't change the field type. For custom phone and email fields, you also can't change the field format.

Important: When you encrypt the Name field, enhanced lookups are automatically enabled. Enhanced lookups improve the user's experience by searching only through records that have been looked up recently, and not all existing records. Switching to enhanced lookups is a one-way change. You can't go back to standard lookups, even if you disable encryption.

You can't use Schema Builder to create an encrypted custom field.

To encrypt custom fields that have the `Unique` or `External ID` attribute, you can only use deterministic encryption.

Unsupported Custom Fields

Some custom fields can't be encrypted.

- Fields on external data objects
- Fields that are used in an account contact relation
- Fields with data translation enabled
- Rich Text Area fields on Knowledge Articles

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

SEE ALSO:

Set Up Field-Level Encryption

Which Files Are Encrypted?

When you enable Shield Platform Encryption for files and attachments, all files and attachments that can be encrypted are encrypted. The body of each file or attachment is encrypted when it's uploaded.

These kinds of files are encrypted when you enable file encryption:

- Files attached to email
- Files attached to feeds
- Files attached to records
- Images included in Rich Text Area fields
- Files on the Content, Libraries, and Files tabs (Salesforce Files, including file previews, and Salesforce CRM Content files)
- Files managed with Salesforce Files Sync and stored in Salesforce
- Files attached to Chatter posts, comments, and the sidebar
- Notes body text using the new Notes tool
- Files attached to Knowledge articles
- Quote PDFs

These file types and attachments aren't encrypted:

- Chatter group photos
- Chatter profile photos
- Documents

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

- Notes previews in the new Notes tool
- Notes and Notes previews in the old Notes tool

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

What Other Data Elements Can I Encrypt?

In addition to standard and custom field data and files, Shield Platform Encryption supports other Salesforce data. You can encrypt CRM Analytics datasets, Chatter fields, fields in the Salesforce B2B Commerce managed package, and more.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Change Data Capture

Change Data Capture provides near-real-time changes of Salesforce records, so you can synchronize corresponding records in an external data store. If a Salesforce record field is encrypted with Shield Platform Encryption, changes to encrypted field values generate change events. You can encrypt these change events by selecting **Encrypt and deliver Change Data Capture events** on the Encryption Policy page in Setup.

Chatter Feed

Encrypted Chatter data includes data in feed posts and comments, questions and answers, link names, and URLs. It also includes poll choices and questions and content from your custom rich publisher apps.

The revision history of encrypted Chatter fields is also encrypted. If you edit or update an encrypted Chatter field, the old information remains encrypted.

Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The database fields on these objects that house encrypted data is visible from the Encryption Statistics page in Setup.

- ChatterExtensionInstance—Payload
- ChatterExtensionInstance—PayloadVersion
- ChatterExtensionInstance—TextRepresentation
- ChatterExtensionInstance—ThumbnailUrl
- ChatterExtensionInstance—Title
- FeedAttachment—Title
- FeedAttachment—Value
- FeedComment—RawCommentBody
- FeedPollChoice—ChoiceBody
- FeedPost—LinkUrl
- FeedPost—RawBody
- FeedPost—Title
- FeedRevision—RawValue

Some fields listed in the Encryption Statistics aren't visible in the UI by the same name. However, they store all encrypted data that's visible in the UI.

Note: Enabling Encryption for Chatter encrypts all eligible Chatter fields. You can't choose to encrypt only some Chatter fields.

CRM Analytics

Encrypts new CRM Analytics datasets.

Note: Data that was in CRM Analytics before encryption was enabled isn't encrypted. If existing data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other existing data, such as CSV data, must be reimported to become encrypted. Although existing data isn't encrypted, it's still accessible and fully functional in its unencrypted state when encryption is enabled.

Data Cloud

Encrypt data at rest in Data Cloud with a customer-managed root key.

Salesforce B2B Commerce

Shield Platform Encryption for B2B Commerce versions 4.10 and later add an extra layer of security to the data your customers enter in Salesforce B2B Commerce ecommerce storefronts. For a list of the supported fields, see Enable Shield Platform Encryption for B2B Commerce for Visualforce Objects.

Search Indexes

When you encrypt search indexes, each file created to store search results is encrypted.

Platform Encryption Q&A

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Here are some frequently asked questions about platform encryption.

What are the hardware and software requirements for using Platform Encryption?

None. The crypto functions run natively on the Salesforce platform. No custom code is required to encrypt or decrypt data.

Must I encrypt all of my data when using Platform Encryption?

No. Not all data is sensitive, so encryption isn't always required. Also, unnecessarily encrypting data can affect performance and functionality.

When I enable Platform Encryption, how are my existing encrypted fields affected?

The Platform Encryption process doesn't affect fields encrypted using Classic Encryption.

What encryption algorithm is used with Platform Encryption?

The Platform Encryption uses symmetric key encryption and a 256-bit Advanced Encryption Standard (AES) algorithm to encrypt field-level data and files stored on the Salesforce platform. Data encryption and decryption occur on the application servers. Encryption is integrated into the Salesforce application so the application knows when data must be encrypted or decrypted. Whether you're accessing data through the user interface or the API, encryption and decryption are handled the same way.

Can I access tenant secrets using the API?

Yes. For example, you can use the API to define an automatic process to rotate the Platform Encryption key regularly. For detailed information, search for TenantSecret in the *Object Reference for Salesforce and Lightning Platform*.

Do data encryption keys held in memory rotate automatically when Salesforce rotates the master secret?

No. While Salesforce rotates the master secret on a per-release basis, customers' data encryption keys aren't impacted. No new data encryption key is derived automatically.

I use Platform Encryption, and the Encrypted checkbox isn't visible when I create or edit an existing custom field. Why?

Only Email, Phone, Text, Text Area, Text Area (Long), Text Area (Rich), Date, Date/Time and URL custom field types are available for encryption.

What happens to existing data if I rotate a tenant secret?

When you generate a new tenant secret, existing encrypted data remains encrypted and accessible as long as the old tenant secret isn't destroyed. New data is encrypted using the new tenant secret. There's no functional difference to the user.

How finely can I control what data is encrypted with Platform Encryption?

For field data, you control which supported standard and custom fields to encrypt. For files and attachments, you control whether encryption is enabled in your organization.

If I enable Platform Encryption, is the format for custom phone, email, and URL fields preserved?

Yes, formats for custom phone, email, and URL fields are preserved when they're encrypted.

Are the Hardware Security Module (HSM) network appliances shared by multiple tenants?

Yes. Key material produced by an HSM is either a per-release secret or a per-tenant secret. Both are required to encrypt your data, so no two tenants have the same data encryption keys.

Do third-party vendors have access to the Hardware Security Modules (HSM)?

No. Salesforce controls access to the HSMs exclusively.

How long are the tenant secret, primary secret, and data encryption keys?

256 bits in length.

Where is my data encryption key stored?

The keys are stored only in memory and never persisted on disk.

Can I manage my keys outside of Salesforce?

Yes. You can store your key outside of Salesforce and have either the External Key Management service or the Cache-Only Key Service fetch it on demand from a key service that you control.

What is the limit for how many keys we can have?

You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Fields and Files (Probabilistic) tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

What if I already have too many active and archived secrets?

If you run into the 50 limit, review your encryption coverage statistics to find our your active key coverage. Choose one or more keys to destroy. Don't destroy any of them until you synchronize the data they encrypt with an active key.

Are keys I store outside of Salesforce part of the 50-key limit?

There is an across-the-board limit of 50 undestroyed keys. This includes keys managed by external services via EKM, BYOK, and the Cache-Only Key service.

How is my organization-specific key generated?

The data encryption keys are derived by a key derivation function (KDF) that combines a primary secret with an organization-specific tenant secret and a randomly generated 128-bit string.

Where are encryption policies defined?

Your organization defines its own policies.

Can I re-encrypt encrypted data?

Yes. You can review your encryption coverage statistics to find our your active key coverage. Then if you want, you can synchronize the encryption of your data with the most recent tenant secret using the Background Encryption Service.

Can a Platform Encryption key be shared across more than one organization?

No. Encryption keys are specific to an organization and can't be shared with other organizations.

Does encrypting fields, files, and attachments with Platform Encryption count against my organization's storage limits?

No. Encryption and decryption do count against your organization's per-transaction Apex limits, but they aren't counted as separate transactions.

If I can see encrypted data, can Salesforce Support representatives also see the data?

Yes, if they have access to the object, record and field.

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a primary secret that Salesforce maintains. By default, we combine these secrets to create your unique data encryption key (DEK). You can also supply your own final DEK. We use your DEK to encrypt data that your users put into Salesforce, and we use it to decrypt data when your authorized users need it.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Encrypting files, fields, and attachments doesn't affect your org's storage limits.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it's a good idea to familiarize yourself with key terminology.

Components Involved in Deriving Keys

Encryption keys are derived with a combination of hardware security modules (HSMs) and key derivation servers.

Differences Between Classic Encryption and Shield Platform Encryption

Shield Platform Encryption offers two paths toward encrypting data: Field-Level Encryption and Database Encryption. Both offer control over key material and encrypt a broader range of data than Classic Encryption. Each Shield Platform Encryption option offers different data coverage, key management options, and support for functionality such as filtering and sorting. Use the comparison table in this article to help you decide which option best meets your encryption requirements.

How Key Material Is Stored

The critical components of the Security Platform Encryption architecture—the KDF secrets, KDF salt, wrapping keys, and DEKs—are secured using a tiered structure that incorporates wrapped keys, signing, and key derivation.

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key (DEK) in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the regional key management server (KMS) to derive the key. The Shield Platform Encryption service then encrypts the data on the application server. If you opt out of key derivation or use either the External Key Management Service or the Cache-Only Key Service, the encryption service applies your customer-supplied data encryption key directly to your data.

Behind the Scenes: The Search Index Encryption Process

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments so that Salesforce can scale operations. Apache Lucene is used for its core library.

How Shield Platform Encryption Works in a Sandbox

Refreshing a sandbox from a production org creates an exact copy of the production org. If Shield Platform Encryption is enabled on the production org, all encryption settings are copied to the sandbox, including tenant secrets created in production.

Why Bring Your Own Key?

Shield Platform Encryption's Bring Your Own Key (BYOK) feature gives you an extra layer of protection if there's unauthorized access to critical data. It can also help you meet the regulatory requirements that come with handling financial, health, or personal data. After you set up your key material, use Shield Platform Encryption as you always do to encrypt data at rest in your Salesforce org.

Why Isn't My Encrypted Data Masked?

If the Shield Platform Encryption service isn't available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don't want some users to see, revisit those users' field-level security settings, record access settings, and object permissions.

Shield Platform Encryption in Hyperforce

Shield Platform Encryption operates in parallel with volume-level encryption. By default, Hyperforce provides volume-level encryption for data at rest. Volume-level encryption protects all the data on a disk with one encryption key, which Salesforce owns and manages. With Shield Platform Encryption, you can encrypt your data in Hyperforce with unique keys that you control and manage.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Salesforce Extensions for Visual Studio Code, Migration Tool, or Postman, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it's a good idea to familiarize yourself with key terminology.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Cache Key Encrypting Key (Cache KEK)

Data encryption keys temporarily reside in the encrypted key cache for deriving final data encryption keys. The cache KEK encrypts these components while they're in the cache.

Data Encryption

The process of applying a cryptographic function to data that results in ciphertext. The Shield Platform Encryption process uses symmetric key encryption, a 256-bit Advanced Encryption Standard (AES) algorithm that uses cipher block chaining (CBC) mode, and a randomized 128-bit initialization vector (IV) to encrypt data stored on the Salesforce Platform. Data encryption and decryption occur on the application servers.

Data Encryption Key (DEK)

Shield Platform Encryption uses DEKs to encrypt and decrypt data. DEKs are derived on the key management servers (KMS). They use key material split between a per-release primary secret and an org-specific tenant secret stored encrypted in the database. The 256-bit derived keys use a key derivation function (KDF) and exist in memory until evicted from the cache. DEKs are sometimes also provided using the External Key Management service by an external key service that you control.

Encrypted Data at Rest

Data that's encrypted when persisted on disk. Salesforce supports encryption for fields stored in the database; documents stored in files, content, libraries, and attachments; search index files; CRM Analytics datasets; and archived data.

Encryption Key Management

All aspects of key management, such as key generation, processes, and storage. Administrators or users who have the Manage Encryption Keys permission can work with Shield Platform Encryption key material.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Hardware Security Module (HSM)

A secure network appliance that provides cryptography processing and key management for authentication. Shield Platform Encryption uses HSMs to generate and store primary and per-release secret material. HSMs also run the key derivation function that derives DEKs used by the encryption service to encrypt and decrypt data. Salesforce uses FIPS 140-2 Level 3 certified HSM devices. HSMs reside within the primary and regional key management servers (KMSs).

High Assurance Virtual Ceremony (HAVC)

A secure meeting among Salesforce Cryptographic officers. During the HAVC, the cryptographic officers convene in secure facilities to generate the per-release secrets material by using the primary HSM. The per-release secrets are then stored within the primary KMS.

Initialization Vector (IV)

Also known as search index. A random sequence used with a key to encrypt data. Shield Platform Encryption IVs are generally 128 bits (16 bytes) in size.

Key Derivation

The process of creating highly secure encryption keys from highly secure key material components. Keys used for encrypting, signing, and decrypting your data, known as the Data Encryption Keys, are derived by using up to 3 cryptographic components: KDF seed, tenant secret, and initialization vector. These components are stored in separate secure locations. A derived key is never stored on disk, which increases its security.

Key Derivation Function (KDF)

The cryptographic algorithm that Shield Platform Encryption uses to generate DEKs. KDFs take as input one or more secrets and a random IV to derive DEKs. Shield Platform Encryption uses Password-based Key Derivation Function 2 (PBKDF2) with HMAC-SHA-256.

Key Rotation

The process of generating a new tenant secret and archiving the previously active one. Active tenant secrets are used for encryption and decryption. Archived ones are used only for decryption until all data has been re-encrypted by using the new, active tenant secret.

Key Wrapping Key (KWK)

A derived symmetric key used to encrypt other keys for secure storage and transport. A primary KWK is used to encrypt the KDF seed, KDF salt, tenant wrapping key, and transit wrapping private key for Transaction Layer Security (TLS) before they're stored in the regional KMS.

Primary HSM

The HSM that resides in the primary key management server (KMS). It generates secure, random secrets for each Salesforce release. The primary HSM is under a strict access protocol and is available to create secrets only through the coordinated actions of multiple trusted cryptographic officers.

Primary Initialization Vector (KDF Salt)

Initialization vector created each release by the primary HSM. It's used in conjunction with organization tenant secrets to derive data encryption keys.

Primary Secret (KDF Seed)

Formerly master secret. Used with the tenant secret and key derivation function to generate a derived data encryption key. (Customers can opt out of key derivation.) The primary secret is rotated each release by using an HSM. No Salesforce employees have access to these keys in cleartext.

Root Key

A key used by Salesforce to secure and control data encryption keys. Root keys can be generated and managed in Salesforce or outside of Salesforce via an external key management service. Depending on the feature and service, data encryption keys controlled by root keys can be customer managed or managed on behalf of the customer by the Shield KMS.

Tenant Secret

An organization-specific secret used in conjunction with the primary secret and key derivation function (KDF) to generate a derived data encryption key (DEK). No Salesforce employees have access to these keys in cleartext.

SEE ALSO:

How Key Material Is Stored

Components Involved in Deriving Keys

Encryption keys are derived with a combination of hardware security modules (HSMs) and key derivation servers.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Application Servers

Servers in production environments that run Salesforce. When a customer attempts to read or write encrypted data or generate a tenant secret, the application server communicates with a regional KMS to process the request.

External Key Management Service

Service that you use when fully managing your own data encryption keys by using the External Key Management Service or the Cache-Only Key Service.

Primary HSM (nShield® Connect HSM model XC)

A FIPS 140-2 Level 3 hardware-compliant network appliance that generates per-release secrets and secret-wrapping keys and signs the public keys of regional HSMs. The primary HSM is located in the primary KMS. Access to the HSM is controlled through a High Assurance Virtual Ceremony (HAVC).

The primary HSM public signing key is used to sign and verify each regional HSM's public encryption key. At the start of each release, the primary and regional HSM public encryption keys are used to separately encrypt a per-release primary key wrapping key, which is used to encrypt the remainder of the per-release secrets used to derive data encryption keys.

Salesforce Search Index

Servers in production environments that manage Salesforce searches. When a user attempts to query encrypted data, the search index processes the request.

Shield KMS Server

Shield Platform Encryption uses a single primary KMS and multiple regional KMSs. The primary KMS is the first KMS to receive the per-release secrets. It makes those secrets available to regional KMSs, and it services key material requests like any regional KMS server.

Differences Between Classic Encryption and Shield Platform Encryption

Shield Platform Encryption offers two paths toward encrypting data: Field-Level Encryption and Database Encryption. Both offer control over key material and encrypt a broader range of data than Classic Encryption. Each Shield Platform Encryption option offers different data coverage, key management options, and support for functionality such as filtering and sorting. Use the comparison table in this article to help you decide which option best meets your encryption requirements.

Feature	Classic Encryption	Field-Level Encryption	Database Encryption
Pricing	Included in base user license	Additional fee applies	Additional fee applies
Encryption at Rest	✓	✓	✓
Native Solution (No Hardware or Software Required)	✓	✓	✓
Encryption Algorithm	128-bit Advanced Encryption Standard (AES)	256-bit Advanced Encryption Standard (AES CBC)	256-bit Advanced Encryption Standard (AES GCM)
HSM-based Key Derivation	✗	✓	✓
Manage Encryption Keys Permission	✗	✓	✓
Generate Keys	✓	✓	✓
Export, Import, and Destroy Keys	✓	✓	✗
Advanced Key Options	✗	BYOK, Cache-only Keys, External Key Management	BYOK
PCI-DSS L1 Compliance	✓	✓	✓
Masking	✓	✗ No (Why Isn't my Encrypted Data Masked?)	✗ No (Why Isn't my Encrypted Data Masked?)
Mask Types and Characters	✓	✗	✗
View Encrypted Data Permission Required to Read Encrypted Field Values	✓	✗	✗

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Feature	Classic Encryption	Field-Level Encryption	Database Encryption
Encrypted Standard Fields	✗	✓ Limited (What Standard Fields Can You Encrypt?)	✓ All standard fields
Encrypted Attachments, Files, and Content	✗	✓	✓
Encrypted Custom Fields	Dedicated custom field type, limited to 175 characters	✓ Limited (What Custom Fields Can You Encrypt?)	✓ All custom fields
Encrypt Existing Fields for Supported Custom Field Types	✗	✓	✓
Encrypt Custom Metadata and Apex	✓	✓	✓
Search, Filters, and Queries	✗	✓ UI, partial search, lookups, and certain SOSL queries on fields encrypted with the deterministic encryption scheme	✓ All SOSL and SOQL queries except on fields also encrypted with field-level encryption
Sorting	✗	✗	✓ Except on fields also encrypted with field-level encryption
Encrypt the Entire Database Including Standard and Custom Fields, Metadata, and Apex	✗	✗	✓
API Access	✓	✓	✓
Available in Workflow Rules and Workflow Field Updates	✗	✓	✓
Available in Approval Process Entry Criteria and Approval Step Criteria	✗	✓	✓

How Key Material Is Stored

The critical components of the Security Platform Encryption architecture—the KDF secrets, KDF salt, wrapping keys, and DEKs—are secured using a tiered structure that incorporates wrapped keys, signing, and key derivation.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

These artifacts, essential participants in the architecture, are stored:

- Securely on disk in the Salesforce Key Management Server (KMS)
- On the Salesforce application server
- In your database as wrapped units (such as a public key)
- In the Data Encryption Key (DEK) cache

Also, these artifacts can be derived as needed from other wrapped artifacts.

The Salesforce encryption key management process ensures that at no time is any security artifact stored unprotected. We use various methods to protect each type of security artifact.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Method	Description
Application Servers	Servers in production environments that run Salesforce. When a customer attempts to read or write encrypted data or generate a tenant secret, the application server communicates with a regional KMS to process the request.
External Key Management Service	Service that you use when fully managing your own data encryption keys by using the External Key Management Service or the Cache-Only Key Service.
Primary HSM (nShield® Connect HSM model XC)	A FIPS 140-2 Level 3 hardware-compliant network appliance that generates per-release secrets and secret-wrapping keys and signs the public keys of regional HSMs. The primary HSM is located in the primary KMS. Access to the HSM is controlled through a High Assurance Virtual Ceremony (HAVC). The primary HSM public signing key is used to sign and verify each regional HSM's public encryption key. At the start of each release, the primary and regional HSM public encryption keys are used to separately encrypt a per-release primary key wrapping key, which is used to encrypt the remainder of the per-release secrets used to derive data encryption keys.
Salesforce Search Index	Servers in production environments that manage Salesforce searches. When a user attempts to query encrypted data, the search index processes the request.
Shield KMS Server	Shield Platform Encryption uses a single primary KMS and multiple regional KMSs. The primary KMS is the first KMS to receive the per-release secrets. It makes those secrets available to regional KMSs, and it services key material requests like any regional KMS server.

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key (DEK) in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the regional key management server (KMS) to derive the key. The Shield Platform Encryption service then encrypts the data on the application server. If you opt out of key derivation or use either the External Key Management Service or the Cache-Only Key Service, the encryption service applies your customer-supplied data encryption key directly to your data.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

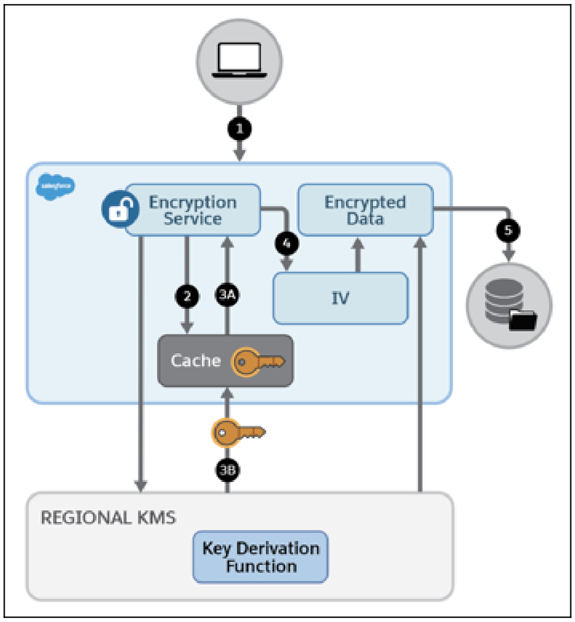
Salesforce securely generates the primary and tenant secrets by using hardware security modules (HSMs). The unique key is derived by using PBKDF2, a key derivation function (KDF), with the primary and tenant secrets as inputs.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Shield Platform Encryption Process Flow



The Shield Platform Encryption process is as follows:

- When a Salesforce user saves encrypted data, the runtime engine determines from metadata whether to encrypt the field, file, or attachment before storing it in the database.
- If so, the encryption service checks for the matching data encryption key in cached memory.
- The encryption service determines whether the key exists.
  - If so, the encryption service retrieves the key.
  - If not, the service sends a derivation request to the regional KMS and returns it to the encryption service running on the Salesforce Platform.

- After retrieving or deriving the key, the encryption service generates a random initialization vector (IV) and encrypts the data by using 256-bit AES encryption.
- The ciphertext is saved in the database or file storage. The IV and corresponding ID of the tenant secret used to derive the data encryption key are saved in the database. Salesforce generates a new primary secret at the start of each release.

Behind the Scenes: The Search Index Encryption Process

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments so that Salesforce can scale operations. Apache Lucene is used for its core library.

Using Shield Platform Encryption's HSM-based key derivation architecture, metadata, and configurations, search index encryption runs when Shield Platform Encryption is in use. The solution applies strong encryption on an org-specific search index (`.fdt`, `.tim`, and `.tip` file types) using an org-specific AES-256 bit encryption key. The search index is encrypted at the search index segment level, and all search index operations require index blocks to be encrypted in memory.

The only way to access the search index or the key cache is through programmatic APIs.

For orgs that use the updated search index framework, search index encryption starts after an admin turns on the option on the Encryption Settings page in Setup. Salesforce creates a root key and DEK. As soon as the DEK is active, search index encryption starts. The admin can turn off search index encryption, generate a new root key, or generate a new DEK. There's no need to configure an encryption policy, because all indexes for all fields are encrypted.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

In orgs that don't yet use the updated search index framework, a Salesforce security administrator can turn on Search Index Encryption from Setup. The administrator first creates a tenant secret of the Search Index type, then they turn on Encryption for Search Indexes. The admin configures their encryption policy by selecting fields and files to encrypt. An org-specific HSM-derived key is derived from the tenant secret on demand. The key material is passed to the search engine's cache on a secure channel.

Note: If Salesforce admins disable encryption on a field, all index segments that were encrypted are unencrypted and the key ID is set to null. This process can take up to seven days.

Process when a user creates or edits records

1. The core application determines whether the search index segment should be encrypted, based on metadata.
2. If the search index segment requires encryption, the encryption service checks for the matching search encryption key ID in the cached memory.
3. The encryption service determines whether the key exists in the cache.
   - If the key exists in the cache, the encryption service uses the key for encryption.
   - If the key doesn't exist in the cache, the service sends a request to the core application, which in turn sends an authenticated derivation request to a key derivation server. The key derivation server then returns the key to the core application server.
4. After retrieving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using NSS or JCE's AES-256 implementation.
5. The key ID (identifier of the key being used to encrypt the index segment) and IV are saved in the search index.

Process when a user searches for encrypted data

1. When a user searches for a term, the term is passed to the search index, along with which Salesforce objects to search.
2. When the search index executes the search, the encryption service opens the relevant segment of the search index in memory and reads the key ID and IV.
3. Steps 3 through 5 of the process when a user creates or edits records are repeated.
4. The search index processes the search and returns the results to the user.

How Shield Platform Encryption Works in a Sandbox

Refreshing a sandbox from a production org creates an exact copy of the production org. If Shield Platform Encryption is enabled on the production org, all encryption settings are copied to the sandbox, including tenant secrets created in production.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

After a sandbox is refreshed, tenant secret changes are confined to your current org. This means that when you rotate or destroy a tenant secret on the sandbox, it doesn't affect the production org.

As a best practice, rotate tenant secrets on sandboxes after a refresh. Rotation ensures that production and sandbox use different tenant secrets. Destroying tenant secrets on a sandbox renders encrypted data unusable in cases of partial or full copies.

Tip: If you use the External Key Management Service, there are special considerations with sandbox key rotation. See External Key Management on page 81.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

SEE ALSO:

EKM in a Sandbox Org

Why Bring Your Own Key?

Shield Platform Encryption's Bring Your Own Key (BYOK) feature gives you an extra layer of protection if there's unauthorized access to critical data. It can also help you meet the regulatory requirements that come with handling financial, health, or personal data. After you set up your key material, use Shield Platform Encryption as you always do to encrypt data at rest in your Salesforce org.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

With Shield Platform Encryption Salesforce administrators can manage the lifecycle of their data encryption keys while protecting these keys from unauthorized access. By controlling the lifecycle of your organization's tenant secrets, you control the lifecycle of the data encryption keys derived from them. Alternatively, you can opt out of key derivation altogether and upload a final data encryption key.

Data encryption keys aren't stored in Salesforce. Instead, they're derived from the primary secret (KDF seed, formerly master secret) and the tenant secret on demand whenever a key is needed to encrypt or decrypt customer data. The primary secret is generated one time per release for everyone

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

during a High Assurance Virtual Ceremony (HAVC) by using a hardware security module (HSM). The tenant secret is unique to your org, and you control when it's generated, activated, revoked, or destroyed.

You have four options for setting up your key material.

- Use Shield Platform Encryption to generate your org-specific tenant secrets.
- Use the infrastructure of your choice, such as an on-premises HSM, to generate and manage your tenant secret outside of Salesforce. Then upload that tenant secret to the regional Salesforce KMS. This option is known as *Bring Your Own Key*, although the element you're really bringing is the tenant secret from which the key is derived.
- Opt out of the Shield Platform Encryption key derivation process with the Bring Your Own Key service. Use the infrastructure of your choice to create a data encryption key instead of a tenant secret. Then upload this data encryption key to the regional Shield KMS. When you opt out of derivation on a key-by-key basis, the Shield Platform Encryption bypasses the derivation process and uses this key material as your final data encryption key. You can rotate customer-supplied data encryption keys just like you can rotate a customer-supplied tenant secret.
- Generate and store your key material outside of Salesforce by using a key service of your choice. Then use either the External Key Management Service or the Salesforce Cache-Only Key Service to fetch your key material on demand. Your key service transmits your key material over a secure channel that you configure. It's then encrypted and stored in the cache for immediate encryption and decryption operations.

SEE ALSO:

Work with External Key Material

Why Isn't My Encrypted Data Masked?

If the Shield Platform Encryption service isn't available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don't want some users to see, revisit those users' field-level security settings, record access settings, and object permissions.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Encryption prevents outsiders from using your Salesforce data even if they manage to get it. It is not a way to hide data from authenticated users. User permissions are the only way to control data visibility for authenticated users. Encryption at rest is about logins, not permissions.

With Shield Platform Encryption, if a user is authorized to see a given set of data, that user sees that data whether it's encrypted or not.

- Authentication means that making sure only legitimate users can get into your system. For example, a company's Salesforce org is only for use by active employees of that company. Anyone who is not an employee is not authenticated; that is, they are barred from logging in. If they do somehow get their hands on the data, it's useless to them because it is encrypted.
- Authorization defines which data or features an authenticated user can use. For example, a sales associate can see and use data in the Leads object, but can't see the regional forecasts, which are intended for sales managers. Both the associate and the manager are properly logged in (authenticated), but their permissions (authorization) are different. That the data is encrypted doesn't make any difference to them.

In general, data can be masked but not encrypted, or encrypted but not masked. For example, regulators often require that only the last four digits of a credit card number be visible to users. Applications typically mask the rest of the number, meaning they replace the digits with asterisks on the user's screen. Without encryption, you can still read the digits that are masked if you can get to the database where they are stored.

Masking might not be enough for your credit card numbers. You may or may not want to encrypt them in the database as well. (You probably should.) If you do, authenticated users will still see the same masked values.

In this way, masking and encryption are different solutions for different problems. You mask data to hide it from users who are authenticated but not authorized to see that data. You encrypt data to prevent someone from stealing the data. (Or, more precisely, to make the data useless if someone does steal it.)

Runtime Masking Notification

If you use Shield Platform Encryption to encrypt fields that you masked, for some fields you can encounter two types of in-field notification instead of the masking value for a field.

- When the field is encrypted but the encryption key has been destroyed
- When either the Shield Platform Encryption or the Masking service is unavailable

If either of these situations occurs, the field displays a value according to the table.

Field Type	Destroyed Key	Service Unavailable
Email, Phone, Text, Text Area, Text Area (Long), URL	?????	!!!!!
Custom Date	08/08/1888	01/01/1777
Custom Date/Time	08/08/1888 12:00 PM	01/01/1777 12:00 PM

Notification values such as ????? and 01/01/1777 are strings reserved for masking notifications and can't be used as data values in encrypted fields. While you aren't restricted from saving a record with one of these reserved masking notification strings into an encrypted field, the field is saved with a blank value. For example, if a Date field is encrypted and you enter 07/07/1777, when you save the record, the contents of that field are empty.

Shield Platform Encryption in Hyperforce

Shield Platform Encryption operates in parallel with volume-level encryption. By default, Hyperforce provides volume-level encryption for data at rest. Volume-level encryption protects all the data on a disk with one encryption key, which Salesforce owns and manages. With Shield Platform Encryption, you can encrypt your data in Hyperforce with unique keys that you control and manage.

Shield Platform Encryption features work in Hyperforce just like they do in implementations running on Salesforce's first-party infrastructure. You can generate a unique key with Salesforce, or bring your own customer-supplied key, and rotate, export, and delete key material on demand. You can also encrypt files and attachments and data in CRM Analytics, Chatter, search indexes, and the event bus. And you can gather statistics about how much of your data is encrypted and, of that data, how much of it's encrypted by active key material. This extra layer of security and control can help you meet your auditing, regulatory, contractual, and compliance requirements.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Salesforce Extensions for Visual Studio Code, Migration Tool, or Postman, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

You can also deploy Shield Platform Encryption using the PlatformEncryptionSettings Metadata API. Regardless of how you deploy, Salesforce automatically checks to see if the implementation violates Shield Platform Encryption guidelines.

Source Organization	Target Organization	Result
Shield Platform Encryption enabled	Shield Platform Encryption enabled	The source Encrypted field attribute indicates enablement.
Shield Platform Encryption enabled	Shield Platform Encryption not enabled	The Encrypted field attribute is ignored.
Shield Platform Encryption not enabled	Shield Platform Encryption enabled	The target Encrypted field attribute indicates enablement.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Set Up Your Encryption Policy

An encryption policy is your plan for encrypting data with Shield Platform Encryption. You can choose how you want to implement it. For example, you can encrypt individual fields and apply different encryption schemes to those fields. Or you can choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption policy.

To provide Shield Platform Encryption for your org, contact your Salesforce account executive. They'll help you provision the correct license so you can create key material and start encrypting data.

Warning: Salesforce recommends testing Shield Platform Encryption in a sandbox org to confirm that your reports, dashboards, processes, and other operations work correctly.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Which User Permissions Does Shield Platform Encryption Require?

Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you do for any other user permission.

Generate and Manage Tenant Secrets

Salesforce has multiple tenant secret types that are used to encrypt different categories of data. You can generate tenant secrets right from Setup.

Set Up Field-Level Encryption

Field-Level Encryption (FLE) gives you fine-grained control over what to encrypt. By encrypting only the specific object fields that contain sensitive information, you can comply with your security needs without undue performance issues. For FLE, we recommend that you encrypt as few fields as necessary. As a Shield Platform Encryption feature, FLE supports custom fields in Lightning Experience, in Salesforce Classic, and in installed managed packages.

Encrypt New Files and Attachments

For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

Encrypt Data in Chatter

Enabling Shield Platform Encryption for Chatter adds an extra layer of security to the information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

Encrypt Data Cloud with Customer-Managed Root Keys

By default, all data in Data Cloud is encrypted at rest in AWS by an AWS-managed data encryption key (DEK). With Platform Encryption for Data Cloud, you can generate a Data Cloud root key in Salesforce. Your Data Cloud root keys are specific to your org and secure the DEKs that encrypt and decrypt your data. In this way, you control the chain of keys that encrypt your data. Generate your Data Cloud root key from Salesforce Setup.

Encrypt Search Index Files with a Tenant Secret

In orgs that don't yet use the updated search index framework, use a tenant secret in the search index encryption process. Sometimes you must search for personally identifiable information (PII) or for data that's encrypted in the database. When you search your org, the results are stored in search index files in plaintext — a potential vulnerability. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

Encrypt Search Index Files with a Root Key

In orgs that use the updated search index framework, you use a DEK that's secured by a root key in the search index encryption process. Sometimes you must search for personally identifiable information (PII) or for data that's encrypted in the database. When you search your org, the results are stored in search index files in plaintext — a potential vulnerability. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

Encrypt CRM Analytics Data

To get started with CRM Analytics Encryption, generate a tenant secret with Shield Platform Encryption. After you generate a CRM Analytics tenant secret, CRM Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your CRM Analytics data.

Encrypt Event Bus Data

To enable encryption of change data capture or platform event messages at rest, generate an event bus tenant secret and then enable encryption.

Fix Compatibility Problems

When you select fields or files to encrypt with Shield Platform Encryption, Salesforce automatically checks for potential side effects. The validation service then warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

Disable Encryption on Fields

You can disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

Which User Permissions Does Shield Platform Encryption Require?

Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you do for any other user permission.

	Manage Encryption Keys	Customize Application	View Setup and Configuration	Manage Certificates
View Platform Encryption Setup pages		✓	✓	
Generate, destroy, export, import, and upload tenant secrets and customer-supplied key material	✓			
Query the TenantSecret object via the API	✓			
Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service	✓	✓		✓
Enable features on the Encryption Settings page	✓	✓		

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

The Customize Application and Manage Certificates permissions are automatically enabled for users with the System Administrator profile.

Restrict Access to Encryption Policy Settings

You can require admins to also have the Manage Encryption Keys permission to complete encryption policy tasks. These tasks include changing the encryption scheme on fields, enabling and disabling encryption on fields, files, and attachments, and other data elements.

To opt in to this feature, you need the Manage Encryption Keys permission. Then opt in from the Encryption Settings page.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
2. In the Advanced Encryption Settings section, turn on **Restrict Access to Encryption Policy Settings**.

   You can also enable Restrict Access to Encryption Policy Settings programmatically. For more information, see PlatformEncryptionSettings in the *Metadata API Developer Guide*.

This restriction applies to actions taken through the API or from Setup pages, such as the Encryption Policy page or the Object Manager.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Generate and Manage Tenant Secrets

Salesforce has multiple tenant secret types that are used to encrypt different categories of data. You can generate tenant secrets right from Setup.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Key Material Types

With Shield Platform Encryption, you encrypt data with either tenant secrets or a key pair composed of a root key and a data encryption key (DEK). Each type of key material targets specific data stores within Salesforce. You can apply different key-rotation cycles or key-destruction policies to different keys based on the kinds of data that they encrypt.

Generate a Tenant Secret with Salesforce

For new customers and admins setting up field-level encryption, generate your first probabilistic and deterministic tenant secrets from the Encryption Settings page. You can also generate any tenant secret from the Key Management page.

Key Material Types

With Shield Platform Encryption, you encrypt data with either tenant secrets or a key pair composed of a root key and a data encryption key (DEK). Each type of key material targets specific data stores within Salesforce. You can apply different key-rotation cycles or key-destruction policies to different keys based on the kinds of data that they encrypt.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Types of Tenant Secrets

Tenant secrets are categorized according to the kind of data that they encrypt.

Fields and Files (Probabilistic)

Encrypts data using the probabilistic encryption scheme, including data in fields, attachments, and files other than search index files

Field (Deterministic)

Encrypts field data by using the deterministic encryption scheme

Search Index

Encrypts fields and other data governed by your encryption policy stored in search indexes. Available in orgs that don't yet use the updated search index framework.

Analytics

Encrypts CRM Analytics data

Event Bus

Encrypts event messages that are stored temporarily in the event bus. For change data capture events, this secret encrypts data changes and the corresponding event that contains them. For platform events, this secret encrypts the event message including event field data.

You can have up to 50 active and archived tenant secrets of each type. For example, you can have 1 active and 49 archived Fields and Files (Probabilistic) tenant secrets and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and key material that you supply.

If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data that it encrypts with an active key.

Root Keys and Data Encryption Keys

Some Salesforce data can be encrypted with a root key and data encryption key (DEK) pair.

AWS Root Key

A root key stored in AWS KMS and referenced by Salesforce, it controls the DEK used to encrypt Salesforce data. Available when External Key Management is enabled, and a connection to AWS KMS is configured.

Salesforce Root Key

Controls the DEK used to encrypt data.

Search Index DEK

Controlled by a root key, it encrypts all search indexes. Available in orgs that use the updated search index framework.

Generate a Tenant Secret with Salesforce

For new customers and admins setting up field-level encryption, generate your first probabilistic and deterministic tenant secrets from the Encryption Settings page. You can also generate any tenant secret from the Key Management page.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys

Generate an Initial Probabilistic or Deterministic Tenant Secret

If you're just getting started with Shield Platform Encryption, you can accomplish a number of your setup tasks on the Encryption Settings page in Setup. Start by turning on settings that generate your first tenant secrets for you. You can then turn on other settings that apply those keys to data, or go to the Encrypt Fields page to apply those tenant secrets to individual fields.

When you turn on settings that generate your first probabilistic and deterministic tenant secret, other settings on the Encryption Settings page become available to you.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Settings**.
2. Turn on one or both of the settings that create an initial tenant secret for you.
   - Turn on **Generate Initial Probabilistic Tenant Secret**. Use the resulting Fields and Files (Probabilistic) tenant secret to encrypt most fields, files, and attachments. This tenant secret must be present before you can generate a deterministic tenant secret.
   - Turn on **Generate Initial Deterministic Tenant Secret**. Use this option to apply the Fields (Deterministic) encryption scheme to fields. This scheme is useful if you want to encrypt fields individually while retaining the ability to sort, filter, and query the contents of those fields.

   Salesforce generates a tenant secret for you. Settings that require an active tenant secret become available on the Encryption Settings page.

With an active tenant secret, you can immediately encrypt custom fields in managed packages or field history and feed tracking values on the Encryption Settings page. You can also go directly to the Encrypt Standard Fields page where you apply tenant secrets to individual fields. See your tenant secrets in the Key Management Table on the Key Management page in Setup.

Create All Tenant Secret Types

New and existing customers can generate tenant secrets of every type on the Key Management page in Setup.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select a key type.
3. Click **Generate Tenant Secret**.

   How often you can generate a tenant secret depends on the tenant secret type. You can generate tenant secrets for the Fields and Files (Probabilistic) type once every 24 hours in production orgs, and once every 4 hours in Sandbox orgs. You can generate tenant secrets for the Search Index type once every 7 days.

   You can have up to 50 active and archived tenant secrets of each type. For example, you can have 1 active and 49 archived Fields and Files (Probabilistic) tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

   If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data that it encrypts with an active key.

Note: This information is about Shield Platform Encryption and not Classic Encryption.

Set Up Field-Level Encryption

Field-Level Encryption (FLE) gives you fine-grained control over what to encrypt. By encrypting only the specific object fields that contain sensitive information, you can comply with your security needs without undue performance issues. For FLE, we recommend that you encrypt as few fields as necessary. As a Shield Platform Encryption feature, FLE supports custom fields in Lightning Experience, in Salesforce Classic, and in installed managed packages.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Shield Platform Encryption supports Field-Level Encryption on standard objects and custom objects. Both standard and custom objects can have standard and custom fields.

After you set up a field for Field-Level Encryption, Shield Platform Encryption begins to encrypt records that are new or that are updated after you enable encryption. To encrypt data that existed before enabling encryption, you can synchronize your existing data with your active key material from the Encryption Statistics and Data Sync page.

There are two ways to configure encryption on object fields. To configure one or more standard fields on any standard object at the same time, you can use the Encrypt Standard Fields page in Setup. To configure encryption for a single standard or custom field, you can also use an object's field details page.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Because the Encrypt Standard Fields page supports only standard fields on standard objects, it doesn't include these fields:

- Custom fields on standard objects
- Standard fields on custom objects
- Custom fields on custom objects

To configure these types of fields for encryption, you must use the standard- or custom-object field details. If a field is eligible for encryption, you can apply it there.

Apply Encryption to Standard Fields in Salesforce Classic

Applying encryption to multiple standard fields at the same time on one or more standard objects is the same process in Salesforce Classic and Lightning Experience. Applying encryption to a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object, in Salesforce Classic is slightly different from the process in Lightning Experience.

Apply Encryption to Standard Fields in Lightning Experience

You can apply encryption to one or more standard fields at the same time on one or more standard objects by using the Encrypt Standard Fields page. To apply encryption to a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object, do one field at a time.

Encrypt Custom Fields in Installed Managed Packages

If an installed managed package supports Shield Platform Encryption, you can encrypt custom fields in that package. Turn on encryption for custom fields in installed managed packages from the Encryption Settings page, and then apply encryption to custom fields in your installed managed package.

SEE ALSO:

Sync Data with Self-Service Background Encryption

Apply Encryption to Standard Fields in Salesforce Classic

Applying encryption to multiple standard fields at the same time on one or more standard objects is the same process in Salesforce Classic and Lightning Experience. Applying encryption to a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object, in Salesforce Classic is slightly different from the process in Lightning Experience.

You can apply encryption to many standard fields at once on one or more standard objects using the Encrypt Standard Fields page. If you need to apply encryption to a custom field on a standard object, or any type of field on a custom object, you do that one field at a time.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt files:

- Customize Application

Apply Encryption to Multiple Standard Fields at the Same Time

You can configure encryption at rest for multiple standard fields across various standard objects at the same time. Use this procedure only for standard fields on standard objects.

To apply deterministic encryption to a standard fields, first turn on deterministic encryption from the Encryption Settings page in Setup.

1. Make sure that your org has an active encryption key. If you're not sure, check with your Salesforce admin.
2. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
3. In the Advanced Encryption Settings section, click **Select Fields**.
   The Encrypt Standard Fields page shows all standard fields for all standard objects.

   Note: This page shows only standard fields on standard objects. Custom fields on standard objects aren't listed. Configure encryption for a custom field from its field details page. Also, configure encryption for an eligible field on a custom object from its field details page.

4. Click **Edit**.
5. Select the fields that you want to encrypt.

By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, from the Encryption Scheme list, select **Deterministic**.

All new data entered in this field is encrypted.

6. Save your work.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings. Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize existing data with your active key material on the Encryption Statistics and Data Sync page.

Apply Encryption to One Standard Field or One Custom Field

Do these steps any time that you want to configure only one field for encryption. This includes a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object.

To apply deterministic encryption to a standard or custom field, first turn on deterministic encryption from the Encryption Settings page in Setup.

Note: This page describes how to apply encryption to a field in Salesforce Classic. To configure a field in Lightning Experience, see Apply Encryption to Standard Fields in Lightning Experience on page 42.

1. From the management settings for the object, go to **Fields**.
2. In the Custom Fields & Relationships section, create a field or edit an existing one.

   If encryption is available for the field, the **Encrypt contents of this field** checkbox appears.

3. Select **Encrypt the contents of this field**.

   By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Advanced Encryption Settings.

   All new data entered in this field is encrypted.

4. Save your work.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings. Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize existing data with your active key material on the Encryption Statistics and Data Sync page.

See Also

- Filter Encrypted Data with Deterministic Encryption
- Sync Data with Self-Service Background Encryption

Apply Encryption to Standard Fields in Lightning Experience

You can apply encryption to one or more standard fields at the same time on one or more standard objects by using the Encrypt Standard Fields page. To apply encryption to a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object, do one field at a time.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt files:

- Customize Application

Apply Encryption to Multiple Standard Fields at the Same Time

You can configure encryption at rest for multiple standard fields across various standard objects at the same time. Use this procedure only for standard fields on standard objects.

To apply deterministic encryption to a standard field, first turn on deterministic encryption from the Encryption Settings page in Setup.

1. Make sure that your org has an active encryption key. If you're not sure, check with your Salesforce admin.
2. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
3. In the Advanced Encryption Settings section, click **Select Fields**.
   The Encrypt Standard Fields page shows all standard fields for all standard objects.

   Note: This page shows only standard fields on standard objects. Custom fields on standard objects aren't listed. Configure encryption for a custom field from its field details page. Also, configure encryption for an eligible field on a custom object from its field details page.

4. Click **Edit**.
5. Select the fields that you want to encrypt.
   All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, from the Encryption Scheme list, select **Deterministic**.
6. Save your work.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings. Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize existing data with your active key material on the Encryption Statistics and Data Sync page.

Apply Encryption to One Standard Field or One Custom Field

Do these steps any time that you want to configure a standard field on a custom object, a custom field on a standard object, or a custom field on a custom object.

To apply deterministic encryption to a standard or custom field, first turn on deterministic encryption from the Encryption Settings page in Setup.

Note: This page describes how to apply encryption to a field in Lightning Experience. To configure encryption for a field in Salesforce Classic, see Apply Encryption to Standard Fields in Salesforce Classic on page 40.

1. From Setup, select **Object Manager**, and then select your object.
2. Click **Fields & Relationships**.

3. When you create or edit a custom field, select **Encrypt the contents of this field**.

   By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Advanced Encryption Settings.

   All new data entered in this field is encrypted.

4. Save your work.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings. Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Synchronize existing data with your active key material on the Encryption Statistics and Data Sync page.

See Also

- Filter Encrypted Data with Deterministic Encryption
- Sync Data with Self-Service Background Encryption

Encrypt Custom Fields in Installed Managed Packages

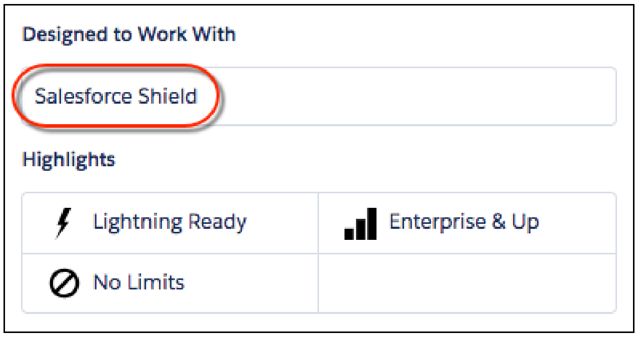
If an installed managed package supports Shield Platform Encryption, you can encrypt custom fields in that package. Turn on encryption for custom fields in installed managed packages from the Encryption Settings page, and then apply encryption to custom fields in your installed managed package.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.

2. In the Advanced Encryption Settings section, turn on **Encrypt Custom Fields in Managed Packages**.

   You can also enable encryption for managed packages programmatically. For more information, see PlatformEncryptionSettings in *Metadata API Developer Guide*.

   From now on, if an installed managed package supports encryption, you can encrypt custom fields in that package. Don't know if your application supports encrypted fields? Look for the Designed to Work With Salesforce Shield marker in your application's AppExchange listing.



Designed to Work With

Salesforce Shield

Highlights

Lightning Ready	Enterprise & Up
No Limits	

If you don't see this marker, talk to your app vendor.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt files:
- Customize Application

Encrypt New Files and Attachments

For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

Note: Before you begin, make sure that your org has an active encryption key. If you're not sure, check with your Salesforce admin.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
2. In the Encryption Policy section, turn on **Encrypt Files and Attachments**.

Important: Users with access to the file can work normally with it regardless of their encryption-specific permissions. Users who are logged in to your org and have read access can search and view the body content.

Users can continue to upload files and attachments per the usual file size limits. Expansion of file sizes caused by encryption doesn't count against these limits.

Turning on file and attachment encryption affects new files and attachments. It doesn't automatically encrypt files and attachments that are already in Salesforce. Apply your active key material to existing data with on the Encryption Statistics and Data Sync page.

To check whether a file or attachment is encrypted, look for the encryption indicator on the detail page of the file or attachment. You can also query the `isEncrypted` field on the ContentVersion object (for files) or on the Attachment object (for attachments).

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

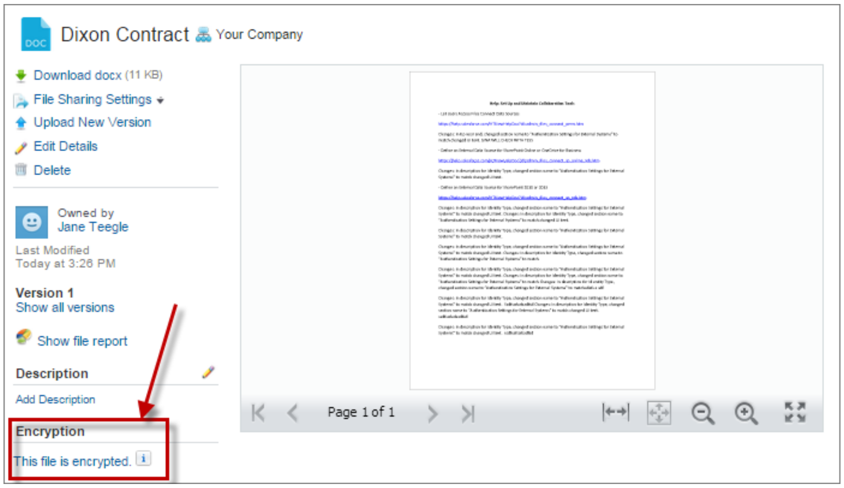
Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt files:
- Customize Application

Here's What It Looks Like When a File Is Encrypted



Note: The encryption indicator is only available in Salesforce Classic.

Encrypt Data in Chatter

Enabling Shield Platform Encryption for Chatter adds an extra layer of security to the information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

We recommend that you test Encryption for Chatter in a dedicated Sandbox environment before enabling it in production.

Unlike encryption for custom and standard fields, enabling encryption for Chatter encrypts all eligible Chatter fields.

1. Make sure that your org has an active encryption key. If you're not sure, check with your administrator.
2. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
3. In the Advanced Encryption Settings section, turn on **Encrypt Chatter**.

The automatic Shield Platform Encryption validation service checks for settings that could block encryption. If the service finds potential problems, it sends you an email with suggestions for fixing the problems.

After you activate encryption for Chatter, new data that you enter into Chatter gets encrypted. To encrypt historic Chatter data, contact Salesforce Customer Support to request the background encryption service.

When you edit or update an encrypted Chatter field, the field's revision history is also encrypted. For example, if you update a post, the old version of the post remains encrypted.

If you enabled Encryption for Chatter in Spring '17 and you want to access the most up-to-date features, deselect **Encrypt Chatter** and then reselect **Encrypt Chatter**.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To encrypt fields:
- Customize Application

Encrypt Data Cloud with Customer-Managed Root Keys

By default, all data in Data Cloud is encrypted at rest in AWS by an AWS-managed data encryption key (DEK). With Platform Encryption for Data Cloud, you can generate a Data Cloud root key in Salesforce. Your Data Cloud root keys are specific to your org and secure the DEKs that encrypt and decrypt your data. In this way, you control the chain of keys that encrypt your data. Generate your Data Cloud root key from Salesforce Setup.

You can generate root keys that encrypt Data Cloud data in both production and sandbox environments.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
2. Turn on **Manage Data Cloud Keys**.
   Salesforce generates a root key for you. When it's ready, you can see it on the Key Management page under the Data Cloud tab.
3. Optionally, you can edit the description on your root for easier key identification and auditing.
   a. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Key Management**.
   b. In the Root Key Inventory section under the Data Cloud tab, click **Details**.
   c. Click **Edit Description**.
   d. Add a unique description, and then save your work.

The latest root key is your active root key. The active root key is used to secure your data encryption keys in AWS, which are used for encrypt and decrypt operations. You can rotate your Salesforce root key for Data Cloud every 3 months. DEKs are generated in AWS as needed.

Your initial DEK is immediately used to encrypt new data in Data Cloud, including search indexes. Salesforce also applies your DEK to existing data, which can take some time if you have a large amount of data in Data Cloud. Check the status of this process on the Data Cloud card on the Encryption Statistics page in Setup.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and Platform Encryption for Data Cloud.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure key material:
- Manage Encryption Keys

To view and edit Setup:
- View Setup and Configuration

Note: Root keys don't control the data encryption keys used to encrypt unstructured data flows in Data Cloud.

Root keys are compatible with Data Cloud's Sub-Second Real-Time feature. When you enable Sub-Second Real-Time in an org with an active Salesforce root key for Data Cloud, the feature can take up to 24 hours to start using that root key.

For Sub-Second Real-Time customers who require customer-managed keys (CMK) encryption in Data Cloud, Salesforce uses tenant level isolation for storing encrypted keys for unified profiles. This isolation ensures that each tenant's data is encrypted with its own keys.

Encrypt Search Index Files with a Tenant Secret

In orgs that don't yet use the updated search index framework, use a tenant secret in the search index encryption process. Sometimes you must search for personally identifiable information (PII) or for data that's encrypted in the database. When you search your org, the results are stored in search index files in plaintext — a potential vulnerability. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

Note: Some orgs use the newer search index encryption functionality. To confirm the encryption type for your org, see Encrypt Search Index Files with a Root Key on page 47.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select **Search Index**.
3. Select **Generate Tenant Secret**.
   This new tenant secret encrypts only the data stored in search index files.
4. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
5. In the Encryption Policy section, turn on **Encrypt Search Indexes**.
   Your search indexes are now encrypted with the active Search Index tenant secret.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys

Encrypt Search Index Files with a Root Key

In orgs that use the updated search index framework, you use a DEK that's secured by a root key in the search index encryption process. Sometimes you must search for personally identifiable information (PII) or for data that's encrypted in the database. When you search your org, the results are stored in search index files in plaintext — a potential vulnerability. You can encrypt these search index files with Shield Platform Encryption, adding another layer of security to your data.

With the Spring '24 release, we began migrating Hyperforce orgs to a new search index encryption architecture. This architecture, available only for Hyperforce orgs, gives you with the ability to control the root key that generates and encrypts the data encryption key (DEK) for your search indexes. The migration is gradual, so it's possible that you're still using the legacy search index encryption. We notify you when your org is using the new architecture.

For orgs that use the updated search index framework, we create the first root key and data encryption key (DEK). Your search indexes are then generated using the new architecture with the new DEK. The old search index tenant secrets are used only until the new search index framework is in place. After your indexes have been reindexed by using the new framework, your old search index tenant secrets are no longer used.

Your search index encryption root key and DEK are both visible on the Key Management page in Setup. The root key that secures a DEK is visible in the Key Management Table. Just like other keys in Salesforce, you can rotate root keys and DEKs for control over your key lifecycle and encryption policy.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys

Search index DEKs are never stored unwrapped. When needed, they're unwrapped by the root key and cached for immediate use by the search index service.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
2. In the Encryption Policy section, turn on **Encrypt Search Indexes**.
   Salesforce begins creating your root key and DEK. You're notified when the new DEK is ready.
3. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
4. In the Key Management Table, select **Search Index**.
   Review the page. When the new DEK is `Active`, your search indexes are being encrypted.

Generate a Search Index Data Encryption Key

In Hyperforce orgs, create the search index encryption data encryption key (DEK) from the Key Management page in Setup. DEKs are secured with Salesforce root keys.

Note: Using Setup is the only way to manage Search Index DEKs. You can't manage them using Apex.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Select the Search Index tab. Then click **Generate DEK**.
   The new DEK is generated. This DEK is used to encrypt all new data in the search index, which builds dynamically as it captures new search data.

   Periodically, more than one iteration of your DEK is needed to encrypt search indexes as they're built. Automatically generated DEK iterations are identifiable by the Automated Process value listed in the Created By column. These iterations of your DEK share a version number.

   When you generate another DEK, all DEKs of the previous version are archived.

EDITIONS

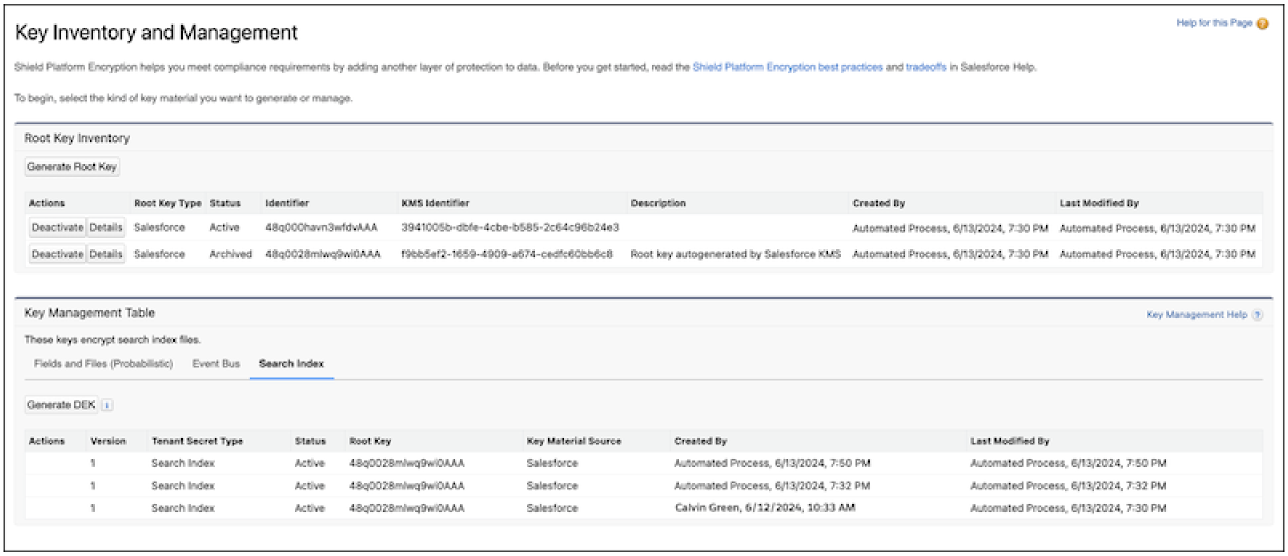
Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available as an add-on subscription to Hyperforce orgs in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys



Key Inventory and Management

Shield Platform Encryption helps you meet compliance requirements by adding another layer of protection to data. Before you get started, read the Shield Platform Encryption best practices and tradeoffs in Salesforce Help.

To begin, select the kind of key material you want to generate or manage.

Root Key Inventory

Actions	Root Key Type	Status	Identifier	KMS Identifier	Description	Created By	Last Modified By
Deactivate Details	Salesforce	Active	48q000havn3wfdvAAA	3941005b-dbfe-4cbe-b585-2c64c96b24e3		Automated Process, 6/13/2024, 7:30 PM	Automated Process, 6/13/2024, 7:30 PM
Deactivate Details	Salesforce	Archived	48q0028mlwq9wi0AAA	f9bb5ef2-1659-4909-a674-cedfc60bb6c8	Root key autogenerated by Salesforce KMS	Automated Process, 6/13/2024, 7:30 PM	Automated Process, 6/13/2024, 7:30 PM

Key Management Table

These keys encrypt search index files.

Actions	Version	Tenant Secret Type	Status	Root Key	Key Material Source	Created By	Last Modified By
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Automated Process, 6/13/2024, 7:50 PM	Automated Process, 6/13/2024, 7:50 PM
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Automated Process, 6/13/2024, 7:32 PM	Automated Process, 6/13/2024, 7:32 PM
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Calvin Green, 6/12/2024, 10:33 AM	Automated Process, 6/13/2024, 7:30 PM

Encrypt CRM Analytics Data

To get started with CRM Analytics Encryption, generate a tenant secret with Shield Platform Encryption. After you generate a CRM Analytics tenant secret, CRM Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your CRM Analytics data.

You must be approved by the CRM Analytics Encryption Product Manager to use CRM Analytics Encryption. To request access, file a case with Salesforce Customer Support.

To learn about CRM Analytic's key management architecture, read Strengthen Your Data's Security with Shield Platform Encryption.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select **Analytics**.
3. Generate a tenant secret or upload key material.
4. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
5. In the Encryption Policy section, select **Encrypt CRM Analytics**.
   New datasets in CRM Analytics are now encrypted.

   Note: Data that was in CRM Analytics before encryption was enabled isn't encrypted. If preexisting data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other preexisting data, such as CSV data, must be reimported to become encrypted. Although preexisting data isn't encrypted, it's still accessible and fully functional in its unencrypted state when encryption is enabled.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing CRM Analytics Platform and either Salesforce Shield or the Platform Encryption add-on.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To manage key material:

- Manage Encryption Keys

Encrypt Event Bus Data

To enable encryption of change data capture or platform event messages at rest, generate an event bus tenant secret and then enable encryption.

These steps enable encryption for change data capture and platform events.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select **Event Bus**.
3. Click **Generate Tenant Secret**, or to upload a customer-supplied tenant secret, click **Bring Your Own Key**, and upload your key.
4. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
5. In the Encryption Policy section, turn on **Encrypt Change Data Capture Events and Platform Events**.

Warning: If you don't enable Shield Platform Encryption for change data capture events and platform events, events are stored in clear text in the event bus.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer**Editions. Requires purchasing either Salesforce Shield or the Platform Encryption add-on.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To manage key material:

- Manage Encryption Keys

Fix Compatibility Problems

When you select fields or files to encrypt with Shield Platform Encryption, Salesforce automatically checks for potential side effects. The validation service then warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

If your results include error messages, you're probably running into one or more of these limitations:

Portals

You can't encrypt standard fields, because a legacy customer or partner portal (created before 2013) is enabled in your organization. To deactivate a legacy customer portal, go to the Customer Portal Settings page in Setup. To deactivate a legacy partner portal, go to the Partners page in Setup.

Note: Experience Cloud sites aren't related to this issue. They're fully compatible with encryption.

Criteria-Based Sharing Rules

You've selected a field that is used in a filter in a criteria-based sharing rule.

SOQL/SOSL queries

You've selected a field that's used in an aggregate function in a SOQL query, or in a WHERE, GROUP BY, or ORDER BY clause.

Formula fields

You've selected a field that's referenced by a custom formula field in an unsupported way. Formulas can use BLANKVALUE, CASE, HYPERLINK, IF, IMAGE, ISBLANK, ISNULL, NULLVALUE, and concatenation (&). Custom formula fields can reference encrypted data in Salesforce Classic but not Lightning Experience or via SOQL.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Flows and Processes

You've selected a field that's used in one of these contexts.

- To filter data in a flow
- To sort data in a flow
- To filter data in a process
- To filter data in a record choice set
- To sort data in a record choice set

Note: By default, your results only list the first 250 errors per element. You can increase the number of errors listed in your results to 5000. Contact Salesforce for help.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Disable Encryption on Fields

You can disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

When you turn off Shield Platform Encryption for a field, most encrypted data is automatically mass-decrypted. The decryption starts automatically after you disable encryption for specific fields and save your changes. When data is decrypted, any functionality that was limited or unavailable when the data was encrypted is also restored. Salesforce notifies you by email when the decryption process is complete.

Note: Automatic decryption takes longer when you disable encryption on fields encrypted with a key that's been destroyed. Salesforce notifies you by email when the process finishes.

Long text area and rich text area field types can't be automatically decrypted. If you decrypt data encrypted with a destroyed key, that data can't be mass-decrypted.

Note: If you disable Shield Platform Encryption and can't access data in fields that were previously encrypted, contact Salesforce for help.

1. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
2. In the Advanced Encryption Settings section, click **Select Fields**.
3. Click **Edit**.
4. Deselect the fields that you want to stop encrypting and save your work.
   Users can see data in these fields.
5. To disable encryption for files and attachments, Chatter, or other data categories, turn off those features from the Encryption Settings page and save your work.

After your data is decrypted, functionality that Shield Platform Encryption limited or changed is restored.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To disable encryption:
- Customize Application

Filter Encrypted Data with Deterministic Encryption

You can filter data that's protected with Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. You can apply case-sensitive deterministic encryption or exact-match case-insensitive deterministic encryption to data on a field-by-field basis.

Deterministic encryption supports WHERE clauses in SOQL queries and is compatible with unique and external ID fields. It also supports single-column indexes and single and double-column unique indexes. Deterministic encryption key types use the Advanced Encryption Standard (AES) with 256-bit keys with CBC mode and a static initialization vector (IV).

How Deterministic Encryption Supports Filtering

By default, Shield Platform Encryption uses a probabilistic encryption scheme to encrypt data. Each bit of data is turned into a fully random ciphertext string every time it's encrypted. Encryption doesn't generally impact users who are authorized to view the data. The exceptions are when logic is executed in the database or when encrypted values are compared to a string or to each other. In these cases, because the data has been turned into random, patternless strings, filtering isn't possible. For example, you might run a SOQL query in custom Apex code against the Contact object, where LastName = 'Smith'. If the LastName field is encrypted with probabilistic encryption, you can't run the query. Deterministic encryption addresses this problem.

To be able to use filters when data is encrypted, we have to allow some patterns in our data. Deterministic encryption uses a static initialization vector (IV) so that encrypted data can be matched to a particular field value. The system can't read a piece of data that's encrypted, but it does know how to retrieve the ciphertext that stands for that piece of data thanks to the static IV. The IV is unique for a given field in a given org and can only be decrypted with your org-specific encryption key.

We evaluate the relative strengths and weaknesses of cryptographic approaches based on the types of attacks that can be launched against a particular algorithm. We also consider the length of time that it could take for the attack to succeed. For example, it is commonly said that a brute-force attack against an AES 256-bit key would take a billion billion years given current computing capabilities. Nevertheless, it is common practice to rotate keys regularly.

Certain kinds of attacks become a bit less far-fetched when you get away from purely random ciphertext. For example, an attacker could conceivably analyze deterministically encrypted ciphertext and determine that the cleartext string `Alice` always resolves to the ciphertext `YjNkY2JlNjU5M2JkNjk4MGJiNWE2NGQ5NzI5MzU1OTcNCg==`. Given enough time to eavesdrop, an attacker could defeat encryption by building a dictionary of cleartext values to ciphertext values.

The Salesforce Shield approach is to expose just enough determinism to let bona fide users filter on encrypted data while limiting it enough to ensure that a given plaintext value doesn't universally result in the same ciphertext value across all fields, objects, or orgs. Even if an attacker successfully matched cleartext to encrypted values for one field, the attacker would have to do it all over again for another field, and again for the same field in another object.

In this way, deterministic encryption decreases encryption strength only as minimally necessary to allow filtering.

Deterministic encryption comes in two types: case-sensitive and case-insensitive. With case-sensitive encryption, a SOQL query against the Contact object, where LastName = Jones, returns only Jones, not jones or JONES. Similarly, when the case-sensitive deterministic scheme tests for unicity (uniqueness), each version of "Jones" is unique.

For case-insensitive, a SOQL query against the Lead object, where Company = Acme, returns Acme, acme, or ACME. When the case-insensitive scheme tests for unicity (uniqueness), each version of Acme is considered identical.

Important: Probabilistic encryption is not supported on the email address field for the Contact object. To avoid creating duplicate accounts during self-registration, use deterministic encryption.

Encrypt Data with the Deterministic Encryption Scheme

Generate key material specific to data encrypted with deterministic encryption schemes. You can apply either case-sensitive deterministic encryption or case-insensitive deterministic encryption schemes to your data, depending on the kind of filtering that you want to perform. When you apply a deterministic encryption scheme to a field or change between deterministic encryption schemes, synchronize your data. Syncing data makes sure that your filters and queries produce accurate results.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

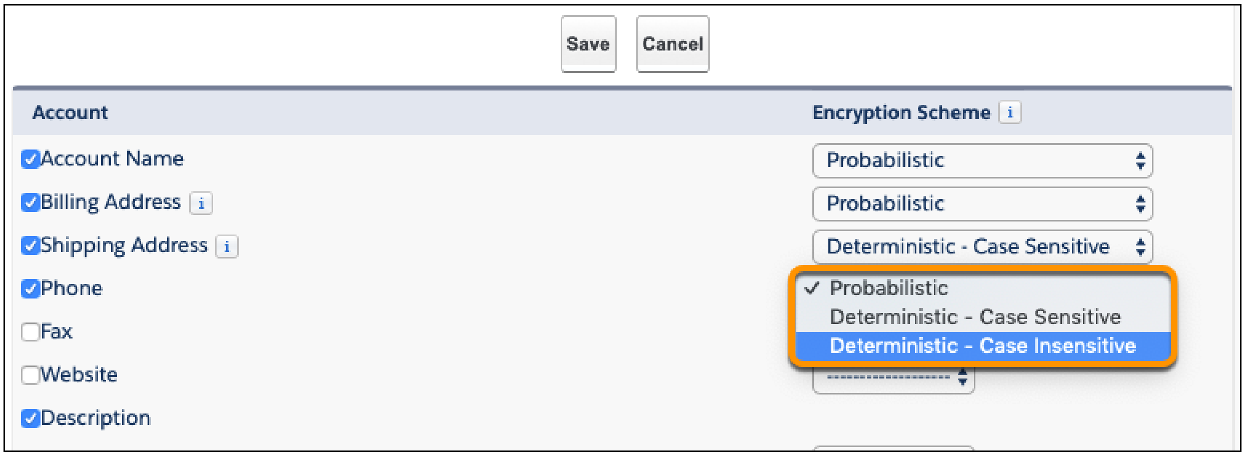
- Manage Encryption Keys

To enable Deterministic Encryption:

- Customize Application

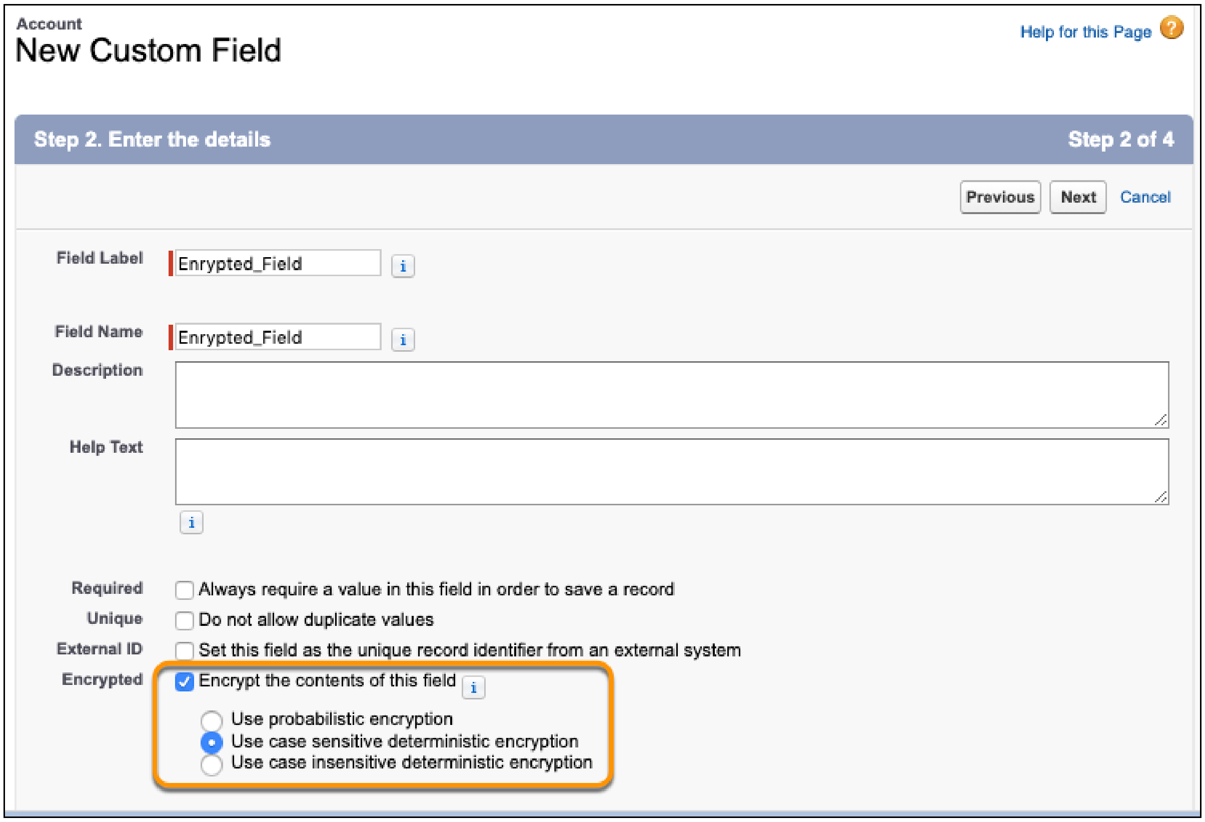
1. If you don't already have an active Fields and Files (Probabilistic) tenant secret, generate one.
    - From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**. Turn on **Generate Initial Probabilistic Tenant Secret**. This path is the fastest because you can stay on the Encryption Settings page to generate your deterministic tenant secret.
    - Optionally, generate this tenant secret on the Key Management page. From Setup, in the Quick Find box, enter **Key Management**, and then select **Key Management**. In the Key Management Table, select **Fields and Files (Probabilistic)**. Then generate or upload a tenant secret.
2. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
3. In the Advanced Encryption Settings section, turn on **Generate Initial Deterministic Tenant Secret**.

    You can also enable deterministic encryption programmatically. For more information, see PlatformEncryptionSettings in the *Metadata API Developer Guide*.
4. Enable encryption for each field, and choose a deterministic encryption scheme. How you do that depends on whether it's a standard field or a custom field.
    - For standard fields, from Setup, select **Encryption Settings**. In the Advanced Encryption Settings section, click **Select Fields**. The Encrypt Standard Fields page opens. For each field that you want to encrypt, select the field name, and then choose either **Deterministic—Case Sensitive** or **Deterministic—Case Insensitive** from the Encryption Scheme list.



Account	Encryption Scheme
☑ Account Name	Probabilistic
☑ Billing Address	Probabilistic
☑ Shipping Address	Deterministic - Case Sensitive
☑ Phone	✓ Probabilistic Deterministic - Case Sensitive Deterministic - Case Insensitive
□ Fax	
□ Website	
☑ Description	

- For custom fields, open the Object Manager and edit the field that you want to encrypt. Select **Encrypt the contents of this field**, and select an encryption scheme.



Account

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Enrypted_Field

Field Name: Enrypted_Field

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

External ID: □ Set this field as the unique record identifier from an external system

Encrypted: ☑ Encrypt the contents of this field

○ Use probabilistic encryption

● Use case sensitive deterministic encryption

○ Use case insensitive deterministic encryption

You can mix and match probabilistic and deterministic encryption, encrypting some fields one way and some fields the other.

You receive an email notifying you when the enablement process finishes.

Note: Expect the enablement process to take longer when you apply deterministic encryption to a field with a large number of records. To support filtering, the enablement process also rebuilds field indexes.

5. When you apply or remove deterministic encryption to a field, it's possible that existing data in that field doesn't appear in queries or filters. To apply full deterministic functionality to existing data, synchronize all your data with your active key material from the Encryption Statistics and Data Sync page. For more information, see Synchronize Your Data Encryption with the Background Encryption Service.

Key Management and Rotation

With Shield Platform Encryption, you control and rotate the key material used to encrypt your data. You can use Salesforce to generate a tenant secret for you, which is then combined with a primary secret for each release to derive a data encryption key. This derived data encryption key is then used in encryption and decryption functions. You can also use the Bring Your Own Key (BYOK) service to upload your own key material. Or you can store your key material outside of Salesforce. Use the External Key Management Service or the Cache-Only Key Service to fetch your key material on demand.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Key management begins with assigning appropriate permissions to security administrators. Assign permissions to people you trust to encrypt data, manage certificates, and work with key material. It's a good idea to monitor these users' key management and encryption activities with the Setup Audit Trail. Authorized developers can generate, rotate, export, destroy, reimport, and upload tenant secrets by coding a call to the TenantSecret object in the Salesforce API.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To manage key material:

- Manage Encryption Keys

Work with Salesforce Key Material

By using Shield Platform Encryption, you can generate a unique tenant secret for your org, or generate a tenant secret or key material using your own external resources. In either case, you manage your own key material: You can rotate it, archive it, and designate other users to share responsibility for it.

Get Statistics About Your Encryption Coverage

The Encryption Statistics page provides an overview of all data encrypted with Shield Platform Encryption. This information helps you to stay on top of your key rotation and management tasks. You can also use encryption statistics to identify which objects and fields you may want to update after you rotate your key material.

Synchronize Your Data Encryption with the Background Encryption Service

Periodically, you change your encryption policy. Or you rotate your keys. To get the most protection out of your encryption strategy with Shield Platform Encryption, synchronize new and existing encrypted data under your most recent encryption policy and keys. You can do this yourself or ask Salesforce for help.

Work with External Key Material

So you can maintain tighter control over your key material, Salesforce offers you three options: BYOK (Bring Your Own Key), EKM (External Key Management), and the Cache-Only key service.

SEE ALSO:

Monitor Setup Changes with Setup Audit Trail

Work with Salesforce Key Material

By using Shield Platform Encryption, you can generate a unique tenant secret for your org, or generate a tenant secret or key material using your own external resources. In either case, you manage your own key material: You can rotate it, archive it, and designate other users to share responsibility for it.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Note: When you generate or upload new key material, it becomes the active key. Any new data is encrypted using this key. However, existing sensitive data remains encrypted using previous keys, which are now archived. In this situation, we strongly recommend re-encrypting this data with your active key. You can synchronize your data with the active key material on the Encryption Statistics and Data Sync.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To manage key material:

- Manage Encryption Keys

Rotate Your Encryption Key Material

You control the lifecycle of your data encryption keys by controlling the lifecycle of your key material. Salesforce recommends that you regularly generate or upload new Shield Platform Encryption key material. When you rotate a tenant secret, data encryption key (DEK), or root key, you replace it with either Salesforce-generated key material or key material that you supply.

Back Up Your Tenant Secrets

Your Shield Platform Encryption tenant secret is unique to your org and to the specific data to which it applies. Salesforce recommends that you export your tenant secret to ensure continued access to the related data.

Destroy Key Material

Only destroy Shield Platform Encryption tenant secrets and key material in extreme cases where access to related data is no longer needed. Your key material is unique to your org and to the specific data to which it applies. Once you destroy key material, related data is not accessible unless you import previously exported key material.

Require Multi-Factor Authentication for Key Management

Multi-factor authentication (MFA) is a powerful tool for securing access to data and resources. Salesforce requires the use of MFA for all logins to your org's user interface. In addition, you can add extra security by also requiring MFA for Shield Platform Encryption key management tasks like generating, rotating, or uploading key material and certificates.

SEE ALSO:

Work with External Key Material

Rotate Your Encryption Key Material

You control the lifecycle of your data encryption keys by controlling the lifecycle of your key material. Salesforce recommends that you regularly generate or upload new Shield Platform Encryption key material. When you rotate a tenant secret, data encryption key (DEK), or root key, you replace it with either Salesforce-generated key material or key material that you supply.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

To decide how often to rotate, consult your security policies. How frequently you can rotate key material depends on the type and environment. For secrets that have restrictions, you can rotate tenant secrets one time per interval.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys

Table 1: Key Material Rotation Intervals

Key Material	Key Type	Production Environments	Sandbox Environments
Fields and Files (Probabilistic)	Tenant secret	24 hours	4 hours
Fields (Deterministic)	Tenant secret	7 days	4 hours
Analytics	Tenant secret	24 hours	4 hours
Event Bus	Tenant secret	7 days	7 days
Search Index	Tenant secret	7 days	7 days
Search Index	DEK	1 hour	1 hour
Salesforce	Root Key	No restriction	No restriction
Salesforce (for Data Cloud data)	Root Key	3 months	3 months

Table 2: Key Material Statuses

Key Type	Key Statuses
AWS Root	Active, Activation Pending, Archived, Canceled, Inactive
Salesforce Root (for Data Cloud data)	Active, Archived
Salesforce Root	Active, Archived, Inactive
Search DEK	Active, Archived, Destroyed
Tenant Secret	Active, Archived, Destroyed

A key's status means the same thing regardless of key type.

Active
The key can be used to encrypt and decrypt new and existing data.

Activation Pending
The key is generated in Salesforce but waiting for another process to complete activation.

Archived
The key can't encrypt new data. It can be used to decrypt data previously encrypted with this key when it was active.

Canceled
The root key activation process is canceled.

Destroyed
The key can't encrypt or decrypt data. Data encrypted with this key when it was active can no longer be decrypted. Files and attachments encrypted with this key can no longer be downloaded.

Inactive

The root key is present but inactive, which prevents DEKs that it controls from encrypting and decrypting data.

Rotate Root Keys and Data Encryption Keys

Shield Platform Encryption encrypts some data stores with key pairs composed of a root key and a data encryption key (DEK). Depending on the data store, you can rotate one or both keys in a key pair. Rotating root keys, which secure DEKs, can help you meet your compliance requirements for key handling. For data stores that allow for customer-managed DEKs, such as search indexes, you can also rotate DEKs. When you rotate a root key, the new root key becomes the active root key. Archived root keys continue to secure existing DEKs. When you rotate a DEK, it's secured by the active root key.

1. From Setup, in the Quick Find box, enter *`Key Management`*, and then select **Key Management**.
2. In the Root Key Inventory, select a root key type tab. Click **Generate Root Key**, and then follow the prompts for generating a new root key.
   The new root key becomes the active root key and is used to secure new DEKs. Archived root keys continue to secure older DEKs that were generated when those root keys were active.
3. In the Key Management Table, select a key type tab. If that key type supports DEKs, you see the option to rotate the DEK. Click **Generate DEK**.
   The new DEK becomes the active DEK. It's secured by the active root key and encrypts new data from that time onward. Archived DEKs continue to decrypt data that they had encrypted. Archived DEKs are secured by the root key that was active when the DEK was generated.

Rotate Tenant Secrets

As with other key material, rotate Shield Platform Encryption tenant secrets to help you stay in alignment with your security and compliance obligations.

The key derivation function uses a primary secret (KDF seed, formerly master secret), which is rotated with each major Salesforce release. Primary secret rotation doesn't affect your encryption keys or your encrypted data until you rotate your tenant secret.

1. From Setup, in the Quick Find box, enter *`Platform Encryption`*, and then select **Key Management**.
2. In the Key Management Table, select a key type.
3. Check the status of the data type's tenant secrets.
4. Click **Generate Tenant Secret** or **Bring Your Own Key**. If you're using a tenant secret of your own, upload your encrypted tenant secret and tenant secret hash.

Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have 1 active and 49 archived Fields and Files (Probabilistic) tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and key material that you supply.

If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

5. If you want to re-encrypt field values with your active key material, synchronize new and existing encrypted data under your most recent and keys. You can sync data from the Encryption Statistics and Data Sync page in Setup.

Back Up Your Tenant Secrets

Your Shield Platform Encryption tenant secret is unique to your org and to the specific data to which it applies. Salesforce recommends that you export your tenant secret to ensure continued access to the related data.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the table that lists your keys, find the tenant secret you want to back up. Click **Export**.
3. Confirm your choice in the warning box, then save your exported file.

   The file name is `tenant-secret-org-<organization ID>-ver-<tenant secret version numer>.txt`. For example, `tenant-secret-org-00DD00000007eTR-ver-1.txt`.
4. Note the specific version you're exporting, and give the exported file a meaningful name. Store the file in a safe location so you can import it back into your org if needed.

   Note: Your exported tenant secret is itself encrypted.

Remember that exported key material is a copy of the key material in your org. To import an exported tenant secret, first destroy the original in your org. See Destroy a Tenant Secret on page 60.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Destroy Key Material

Only destroy Shield Platform Encryption tenant secrets and key material in extreme cases where access to related data is no longer needed. Your key material is unique to your org and to the specific data to which it applies. Once you destroy key material, related data is not accessible unless you import previously exported key material.

You are solely responsible for making sure that your data and key material are backed up and stored in a safe place. Salesforce can't help you with deleted, destroyed, or misplaced tenant secrets and keys.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the table that lists your tenant secrets, find the row that contains the one you want to destroy. Click **Destroy**.
3. A warning box appears. Type in the text as shown and select the checkbox acknowledging that you're destroying a tenant secret, then click **Destroy**.
   After you destroy the key that encrypted the content, file previews and content that was already cached in the user's browser may still be visible in cleartext. When the user logs in again, the cached content is removed.

   If you create a sandbox org from your production org and then destroy the tenant secret in your sandbox org, the tenant secret still exists in the production org.
4. To import your tenant secret, click **Import** > **Choose File** and select your file. Make sure you're importing the correct version of the tenant secret.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:
- Manage Encryption Keys

Require Multi-Factor Authentication for Key Management

Multi-factor authentication (MFA) is a powerful tool for securing access to data and resources. Salesforce requires the use of MFA for all logins to your org's user interface. In addition, you can add extra security by also requiring MFA for Shield Platform Encryption key management tasks like generating, rotating, or uploading key material and certificates.

Important: Make sure that you provide security administrators a way to get a time-based, one-time password. This password is their second authentication factor (in addition to their Salesforce username and password). Otherwise, they can't complete encryption key-related tasks.

1. From Setup, in the Quick Find box, enter `Identity Verification`, and then select **Identity Verification**.
2. Select **Raise session to high-assurance** from the Manage Encryption Keys dropdown.
   All admins with the Manage Encryption Keys permission must use an additional verification method to complete key management tasks through Setup and the API.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign identity verification for key management tasks:
- Manage Encryption Keys

Get Statistics About Your Encryption Coverage

The Encryption Statistics page provides an overview of all data encrypted with Shield Platform Encryption. This information helps you to stay on top of your key rotation and management tasks. You can also use encryption statistics to identify which objects and fields you may want to update after you rotate your key material.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Gather Encryption Statistics

The Encryption Statistics and Data Sync page shows you how much of your data is encrypted by Shield Platform Encryption, and how much of that data is encrypted by active key material. Use this information to inform your key rotation actions and timelines. You can also use the Encryption Statistics page to collect information about the fields and objects you want to synchronize with the background encryption service.

Interpret and Use Encryption Statistics

The Encryption Statistics page offers a snapshot of your encrypted data. You can use the information to help you make informed decisions about managing your encrypted data.

Gather Encryption Statistics

The Encryption Statistics and Data Sync page shows you how much of your data is encrypted by Shield Platform Encryption, and how much of that data is encrypted by active key material. Use this information to inform your key rotation actions and timelines. You can also use the Encryption Statistics page to collect information about the fields and objects you want to synchronize with the background encryption service.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Statistics**.
2. Select an object type or custom object from the left pane. If you see a "--" in the Data Encrypted or Uses Active Key columns, you haven't gathered statistics for that object yet.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

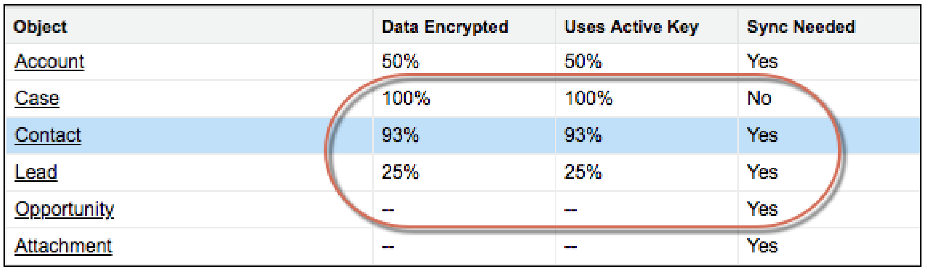
USER PERMISSIONS

To view Platform Encryption Setup pages:

- View Setup and Configuration

  And

  Customize Application



Object	Data Encrypted	Uses Active Key	Sync Needed
Account	50%	50%	Yes
Case	100%	100%	No
Contact	93%	93%	Yes
Lead	25%	25%	Yes
Opportunity	–	–	Yes
Attachment	–	–	Yes

3. Click **Gather Statistics**.
   The gathering process time varies depending on how much data you have in your object. You're notified by email when the gathering process is finished. When your statistics are gathered, the page shows updated information about data for each object. If encryption for field history and feed tracking is turned on, you also see stats about encrypted field history and feed tracking changes.

Note:

- You can gather statistics once every 24 hours, either by clicking **Gather Statistics** or running the self-service background encryption service.
- Feed Item doesn't display statistics because it's derived from Feed Post. Gathering statistics for Feed Post is sufficient to confirm the encryption status of both Feed Post and Feed Item.

Interpret and Use Encryption Statistics

The Encryption Statistics page offers a snapshot of your encrypted data. You can use the information to help you make informed decisions about managing your encrypted data.

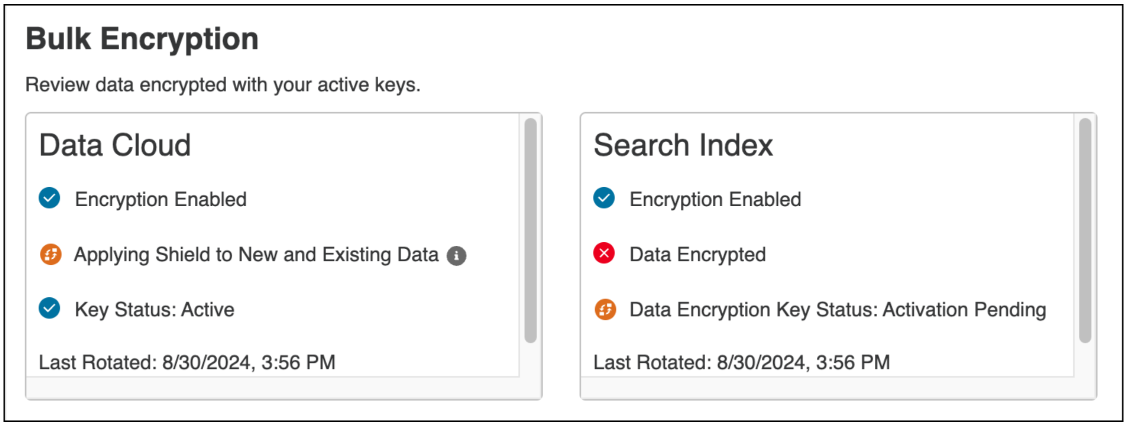
Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge.

Available in both Salesforce Classic and Lightning Experience.

The page offers three views of your encrypted data: summary cards for encrypted data categories, a field-level encryption summary panel, and an encrypted field detail view.

Summary Cards

Shield Platform Encryption encrypts some compatible databases in bulk, such as search indexes and Data Cloud. Summary cards show encryption statistics for these databases, including whether encryption is enabled for that category of data and if that data is encrypted. When an encryption key is present, the summary cards also show the status of that key and when it was last rotated.



Bulk Encryption

Review data encrypted with your active keys.

Data Cloud

✓ Encryption Enabled

⟳ Applying Shield to New and Existing Data ⓘ

✓ Key Status: Active

Last Rotated: 8/30/2024, 3:56 PM

Search Index

✓ Encryption Enabled

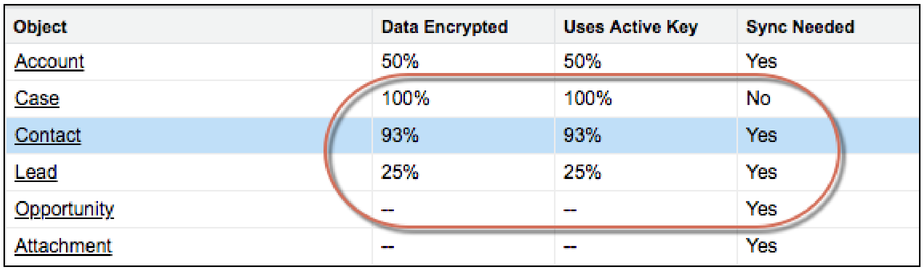
✗ Data Encrypted

⟳ Data Encryption Key Status: Activation Pending

Last Rotated: 8/30/2024, 3:56 PM

Field-Level Encryption Summary View

The Encryption Summary View lists all your objects that contain encrypted data and statistics about the encrypted data in those objects.



Object	Data Encrypted	Uses Active Key	Sync Needed
Account	50%	50%	Yes
Case	100%	100%	No
Contact	93%	93%	Yes
Lead	25%	25%	Yes
Opportunity	--	--	Yes
Attachment	--	--	Yes

- Object—Lists your standard and custom objects. Data about standard objects are aggregated for all standard objects of a given type. Data about custom objects are listed for each custom object.
- Data Encrypted—The total percentage of data in an object that's encrypted. In the example above, 50% of all data in Account objects is encrypted.
- Uses Active Key—The percentage of your encrypted data in that object or object type that's encrypted with your active key material.
- Sync Needed—Recommends whether to synchronize your data with the background encryption service. This column displays Yes when you add or disable encryption on fields, change a field's encryption scheme, or rotate key material.

When the numbers in the Data Encrypted and Uses Active Key columns are the same, and the Sync Needed column is No, all your encrypted data is synchronized. In the example above, the Case object is synchronized.

Sometimes the Sync Needed column is Yes for an object when the Encrypted Data and Uses Active Key columns have the same values. This combination of values happens when encryption policy settings or keys change since the last time that you gathered statistics or synchronized your data. This combination also happens when statistics are gathered for newly encrypted data but the object hasn't been synchronized. In the example above, the Account, Contact, Lead, and Opportunity objects meet one or more of these conditions.

A double dash (--) means that statistics haven't been gathered for that object or object type yet. In the example, statistics haven't been gathered for the Opportunity and Attachment objects.

Encryption Detail View

The Encryption Detail View shows statistics about the field and historical data stored in each object category. If encryption for field history and feed tracking is turned on, you can also view stats about encrypted field history and feed tracking changes.

Fields

The Fields tab displays data about field data in each object.

- Field—All encryptable standard and custom fields in the object that contain data

  Note: Not all field data is stored in the same field that displays data in the UI. For example, some Person Account field data is stored in the corresponding Contact fields. If you have Person Accounts enabled but don't see encrypted fields under the Account detail view, gather statistics for the Contact object and check there.

  Similarly, Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The Encryption Statistics page lists these objects and all fields that hold encrypted Chatter data in the database. Some fields listed on the Encryption Statistics page aren't visible in the UI by the same name, but they store all encrypted data that's visible in the UI. See Which Standard Fields Can I Encrypt? in *Salesforce Help* for a list of the encrypted Chatter fields.

- API Name—The API name for fields that contain data.
- Encrypted Records—The number of encrypted values stored in a field type across all objects of a given type. For example, you select the Account object and see "9" in the Encrypted Records column next to Account Name. That means there are nine encrypted records across all Account Name fields.
- Unencrypted Records—The number of plaintext values stored in a field type.
- Mixed Tenant Secret Status—Indicates whether a mixture of active and archived tenant secrets apply to encrypted data in a field type.
- Mixed Schemes— Indicates whether a mixture of deterministic and probabilistic encryption schemes apply to encrypted data in a field type.

Note: For encrypted and unencrypted records:

- The records count for a field doesn't include NULL or BLANK values. A field with NULL or BLANK values can show a different (smaller) records count than the actual number of records.
- The records count for compound fields such as Contact.Name or Contact.Address can show a different (larger) records count than the actual number of records. The count includes the two or more fields that are counted for every record.

History

The History tab shows data about field history and feed tracking changes.

- Field—All encryptable standard and custom fields in the object that contain data.
- API Name—The API name for fields that contain data.
- Encrypted Field History—The number of encrypted field history values for a field type across all objects of a given type. For example, you select the Account object and see "2" in the Encrypted Field History column for Account Name, which means that Account Name has two encrypted field history values.
- Unencrypted Field History—The number of plaintext field history values stored for a field.
- Encrypted Feed Tracking—The number of encrypted feed tracking values stored for a field.
- Unencrypted Feed Tracking—The number of plaintext feed tracking values stored for a field.

Usage Best Practices

Use these statistics to make informed decisions about your key management tasks.

- Update encryption policies—The encryption statistics detail view shows you which fields in an object contain encrypted data. Use this information to periodically evaluate whether your encryption policies match your organization's encryption strategy.

- Rotate keys—To encrypt all your data with your active key material, review the encryption summary pane on the left side of the page. If the Uses Active Key value is lower than the Data Encrypted value, some of your data uses archived key material. To synchronize your data, click the **Sync** button or contact Salesforce Customer Support.
- Synchronize data—Key rotation is an important part of any encryption strategy. When you rotate your key material, apply the active key material to existing data. To synchronize your data with your active key, click the **Sync** button.

  If self-service background encryption is unavailable, review the Uses Active Key and Mixed Tenant Secret Status columns to identify any fields that include data encrypted with an archived key. Make a note of these objects and fields, then contact Salesforce Customer Support to request the background encryption service. Salesforce Customer Support can focus just on those objects and fields that you want to synchronize, keeping the background encryption process as short as possible.

Synchronize Your Data Encryption with the Background Encryption Service

Periodically, you change your encryption policy. Or you rotate your keys. To get the most protection out of your encryption strategy with Shield Platform Encryption, synchronize new and existing encrypted data under your most recent encryption policy and keys. You can do this yourself or ask Salesforce for help.

When a change occurs, you have options for keeping your encryption policy up to date. You can synchronize most standard and custom field data yourself from the Encryption Statistics and Data Sync page in Setup. For all other data, Salesforce is here to help ensure data alignment with your latest encryption policy and tenant secret.

When We Do and Don't Automatically Encrypt Your Data

- When you turn on encryption for specific fields or other data, newly created and edited data are automatically encrypted with the most recent key.
- Data that's already in your org doesn't automatically get encrypted. Our background encryption service takes care of that on request.
- When you change your tenant secret as part of your key rotation strategy, data that's already encrypted remains encrypted with the old tenant secret. Our background encryption service can update it on request. And don't worry, you always have access to your data as long as you don't destroy the old, archived keys.
- If you turn off encryption, data that's already there is automatically decrypted based on the relevant key. Any functionality impacted by having encrypted data is restored.
- If Salesforce support re-encrypts your data with a new key, any data that was encrypted with the destroyed key is skipped. To access data encrypted with a destroyed key, import a backup of the destroyed key.

Note: Note: Synchronizing your data encryption doesn't modify the record LastModifiedDate or LastModifiedById timestamps. It doesn't execute triggers, validation rules, workflow rules, or any other automated service. However, it does modify the SystemModStamp.

What You Can Synchronize Yourself

You can synchronize most encrypted data yourself from the Encryption Statistics page in Setup. Self-service background encryption synchronizes:

- Standard and custom fields
- The Attachment—Content Body field
- Field history and feed tracking changes when the **Encrypt Field History and Feed Tracking Values** setting is turned on

Read more about self-service background encryption on page 67, and its considerations on page 121, in Salesforce Help.

How to Request Background Encryption Service from Salesforce Customer Support

If you can't sync data yourself, contact Salesforce Customer Support for help. Keep these tips in mind when asking for help with syncing your data.

Allow lead time

Contact Salesforce support 2–3 business days before you need the background encryption completed. The time to complete the process varies based on the volume of data. It could take several days.

Specify the data

Provide the list of objects, field names, and data elements you want encrypted or re-encrypted.

Verify the list

Verify that this list matches what's encrypted in Setup:

- Data elements selected on the Encryption Policy page
- Standard fields selected on the Encrypt Standard Fields page
- Custom fields you selected for encryption on the Field Definition page

Tip: Also check that your field values aren't too long for encryption.

Include files and attachments?

Encryption for files and attachments is all or nothing. You don't have to specify which ones.

Include history and feed data?

Specify whether you want the corresponding field history and feed data encrypted.

Choose a time

Salesforce Customer Support can run the background encryption service Monday through Friday between 6 AM and 5 PM in your time zone.

Tip: If you're not sure which data is already encrypted, visit the Encryption Statistics page, which keeps a record of all fields that you have encrypted.

What If You Destroyed Your Key?

If your encryption key has been destroyed, your data can't be automatically decrypted. You have some options for handling this data.

- Reimport the destroyed key from a backup, then ask Salesforce Customer Support to synchronize your data with your encryption policy.
- Delete all the data that was encrypted with the destroyed key, then ask Salesforce Customer Support to synchronize your data.
- Ask Salesforce Customer Support to mass overwrite the data that was encrypted with the destroyed key with "?????".

Note: Keep these points in mind when disabling encryption on data encrypted with destroyed material.

- When you disable encryption for files that were encrypted with a key that's been destroyed, the files don't automatically go away. You can ask Salesforce support to delete the files.
- The automatic decryption process takes longer when you disable encryption on fields encrypted with a key that's been destroyed. Salesforce notifies you by email when the process finishes.

Sync Data with Self-Service Background Encryption

Synchronizing your data with your active key material keeps your encryption policy up to date. You can sync data in standard and custom fields, the Attachment—Content Body field, and for field history and feed tracking changes from the Encryption Statistics and Data Sync page in Setup. To synchronize all other encrypted data, contact Salesforce Customer Support.

Sync Data with Self-Service Background Encryption

Synchronizing your data with your active key material keeps your encryption policy up to date. You can sync data in standard and custom fields, the Attachment—Content Body field, and for field history and feed tracking changes from the Encryption Statistics and Data Sync page in Setup. To synchronize all other encrypted data, contact Salesforce Customer Support.

Self-service background encryption supports all standard and custom fields, the Attachment—Content Body field, and field history and feed tracking changes. For help synchronizing other encrypted data, contact Salesforce Customer Support.

To include field history and feed tracking values in self-service background encryption processes, first turn on **Encrypt Field History and Feed Tracking Values** on the Encryption Settings page. You can also enable field history and feed tracking encryption programmatically with the PlatformEncryptionSettings metadata type. When this setting is turned on, the self-service background encryption process applies your active key material to your field history and feed tracking values.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Encryption Statistics**.
2. Select an object type or custom object from the left pane.

   Note: The Sync Needed column indicates when to synchronize your data. This column displays Yes when you add or remove encryption on fields, rotate keys, or change a field's encryption scheme.

3. Click **Sync**.
   Supported standard and custom fields are encrypted with your active key material and encryption policy in the background. After the service syncs your data, it gathers statistics for the object. To view your gathered statistics, wait for your verification email and then refresh the Encryption Statistics and Data Sync page.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

View Platform Encryption Setup pages:
- View Setup and Configuration

Note: The sync process time varies depending on how much data you have in your object. You get an email notification when the sync process finishes. You can sync your data from the Encryption Statistics and Data Sync page once every 7 days.

If you have lots of data in Attachment—Content Body fields, the sync process breaks your request into batches and syncs them in sequence. However, sometimes we can't encrypt all these batches at once. This service protection helps Salesforce maintain functional network loads. If the sync process finishes but the encryption statistics status is less than 100% complete, click **Sync** again. The background encryption service picks up where it left off.

Work with External Key Material

So you can maintain tighter control over your key material, Salesforce offers you three options: BYOK (Bring Your Own Key), EKM (External Key Management), and the Cache-Only key service.

Bring Your Own Key (BYOK)

When you supply your own tenant secret or data encryption key (DEK), you get the benefits built into to Salesforce Shield Platform Encryption, plus the extra assurance that comes from exclusively managing your own key material.

External Key Management

Shield External Key Management (EKM) connects your Salesforce implementation to your keys in AWS KMS and uses those keys for encryption operations on Salesforce data. EKM fetches your keys on demand from AWS KMS over a secure channel. EKM stores your key in the key cache and uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cached EKM keys in any system of record or backups. You can revoke key material at any time.

Cache-Only Key Service

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce in any key repository or service that you control and have the Cache-Only Key Service fetch your key on demand from that key service. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

Configure Your Cache-Only Key Callout Connection

Use a named credential to specify the endpoint for your callout, and identify the key that you want to fetch from your endpoint.

SEE ALSO:

Work with Salesforce Key Material

Cache-Only Key Service

Bring Your Own Key (BYOK)

When you supply your own tenant secret or data encryption key (DEK), you get the benefits built into to Salesforce Shield Platform Encryption, plus the extra assurance that comes from exclusively managing your own key material.

Controlling your own tenant secret or DEK entails:

- Contacting Salesforce Customer Support to enable Bring Your Own Keys
- Generating a BYOK-compatible certificate for the type of encryption
- Using that BYOK-compatible certificate to encrypt and secure your self-generated tenant secret or DEK
- Granting the Salesforce Shield Platform Encryption key management machinery access to your tenant secret.

BYOK supports derived keys and DEKs.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

To edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service:

- Manage Encryption Keys

  AND

  Manage Certificates

  AND

  Customize Application

Bring Your Own Key Overview

Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

Generate a BYOK-Compatible Certificate

To encrypt data in Salesforce with Bring Your Own Key (BYOK) key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

Generate and Wrap BYOK Key Material

Generate a random number as your BYOK tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the BYOK-compatible certificate you generated.

Sample Script for Generating a BYOK Tenant Secret

We've provided a helper script that may be handy for preparing your tenant secret for upload. The script generates a random number as your tenant secret, calculates an SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

Upload Your BYOK Key Material

You can provide two types of your own key material for BYOK; tenant secrets, and DEKs. After you create your BYOK-compatible key material, upload it to Salesforce. The process for uploading tenant secrets and DEKs are slightly different. This topic shows you how to do both.

Opt Out of Key Derivation with BYOK

If you don't want Shield Platform Encryption to derive a data encryption key for you, you can opt out of key derivation and upload your own DEK. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

Take Good Care of Your BYOK Keys

When you create and store your own key material outside of Salesforce, it's important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

Troubleshooting Bring Your Own Key

Read these frequently asked questions to help you troubleshoot any problems that arise with Shield Platform Encryption's Bring Your Own Key service.

Bring Your Own Key Overview

Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

To work with our key management machinery, your customer-supplied key material must meet these specifications:

- 256-bit size
- Encrypted with a public 4096-bit RSA key that is extracted from the downloaded BYOK certificate, then padded using the SHA1 padding algorithm with OAEP padding. When you prepare a search index data encryption key or transactional database tenant secret, use SHA512.
- After it's encrypted, it must be encoded in standard base64

To work with encryption keys, you need the Manage Encryption Keys permission. To generate BYOK-compatible certificates, you need the Customize Application permission.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Generate a BYOK-Compatible Certificate

To encrypt data in Salesforce with Bring Your Own Key (BYOK) key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

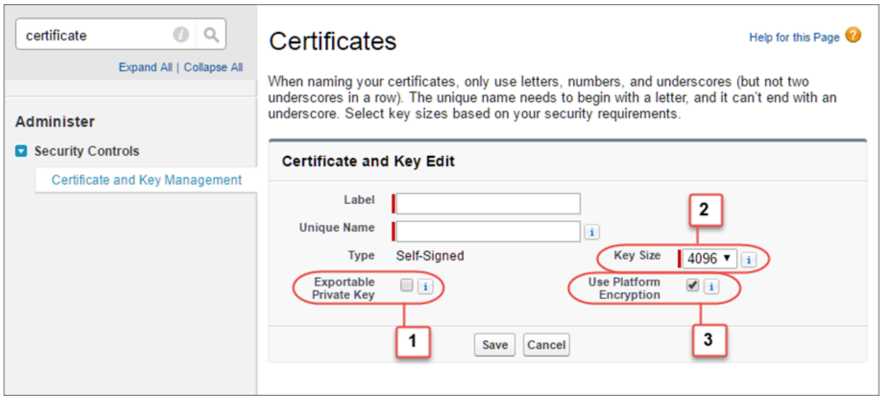
This task shows how to create a self-signed certificate using Setup. If you're not sure whether a self-signed or CA-signed certificate is right for you, consult your organization's security policy. For more information about what each option implies, see Certificates and Keys.

To create a CA-signed certificate, follow the instructions in Generate a Certificate Signed By a Certificate Authority. To make sure that your certificate is BYOK-compatible, remember to manually change the Exportable Private Key, Key Size, and Platform Encryption settings.

To create a self-signed certificate:

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Click **Bring Your Own Key**.
3. Click **Create Self-Signed Certificate**.
4. Enter a unique name for your certificate in the Label field. The Unique Name field automatically assigns a name based on what you enter in the Label field.

   The Exportable Private Key (1), Key Size (2), and Use Platform Encryption (3) settings are preset. (For a BYOK certificate, you must select 4096 for the key size). These settings ensure that your self-signed certificate is compatible with Salesforce Shield Platform Encryption.



Certificates

When naming your certificates, only use letters, numbers, and underscores (but not two underscores in a row). The unique name needs to begin with a letter, and it can't end with an underscore. Select key sizes based on your security requirements.

Certificate and Key Edit

Label

Unique Name

Type Self-Signed

Key Size 4096

Exportable Private Key ☐

Use Platform Encryption ☑

5. When the Certificate and Key Detail page appears, click **Download Certificate**.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service

- Manage Certificates

  AND

  Customize Application

  AND

  Manage Encryption Keys

Generate and Wrap BYOK Key Material

Generate a random number as your BYOK tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the BYOK-compatible certificate you generated.

Note: You can use a tenant secret as a BYOK key only one time. If you need multiple BYOK keys, you need to use a unique tenant secret for each one.

1. Generate a 256-bit tenant secret using the method of your choice.

   You can generate your tenant secret in one of 2 ways:

   - Use your own on-premises resources to generate a tenant secret programmatically, using an open-source library such as Bouncy Castle or OpenSSL.

     Tip: We've provided a script on page 72 that may be useful as a guide to the process.

   - Use a key brokering partner that can generate, secure, and share access to your tenant secret.

2. Wrap your tenant secret with the public key from the BYOK-compatible certificate you generated, using the SHA512 padding algorithm.

   Specify the OAEP padding scheme. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64.

   Note: For legacy BYOK (those not used for tenant secrets, such as BYOK for Search Index encryption and Database Encryption), you can still use the SHA1 padding algorithm.

3. Encode this encrypted tenant secret to base64.
4. Calculate an SHA-256 hash of the plaintext tenant secret.
5. Encode the SHA-256 hash of the plaintext tenant secret to base64.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service:

- Manage Certificates

  AND

  Customize Application

  AND

  Manage Encryption Keys

Sample Script for Generating a BYOK Tenant Secret

We've provided a helper script that may be handy for preparing your tenant secret for upload. The script generates a random number as your tenant secret, calculates an SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

Note: You can use a tenant secret as a BYOK key only one time. If you need multiple BYOK keys, you need to use a unique tenant secret for each one.

1. Download the script from the Salesforce Knowledge Base. Save it in the same directory as the certificate.
2. Run the script specifying the certificate name, like this: *`./secretgen.sh my_certificate.crt`*

   Replace this certificate name with the actual filename of the certificate you downloaded.

   Tip: If needed, use `chmod +w secretgen.sh` to make sure that you have write permission to the file and use `chmod 775` to make it executable.

3. The script generates several files. Look for the two files that end with the .b64 suffix.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

The files ending in .b64 are your base 64-encoded encrypted tenant secret and base 64-encoded hash of the plaintext tenant secret. You'll need both of these files for the next step.

Upload Your BYOK Key Material

You can provide two types of your own key material for BYOK; tenant secrets, and DEKs. After you create your BYOK-compatible key material, upload it to Salesforce. The process for uploading tenant secrets and DEKs are slightly different. This topic shows you how to do both.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

SEE ALSO:

How Key Material Is Stored

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

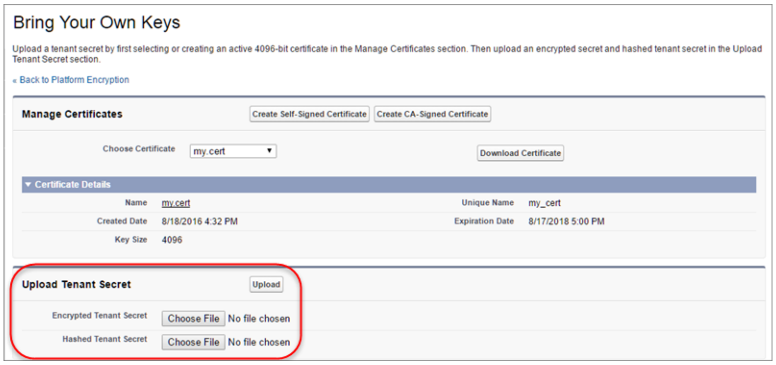
Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

Upload Your BYOK Tenant Secret

After you have your BYOK-compatible tenant secret, upload it to Salesforce. The Shield Key Management Service (KMS) uses your tenant secret to derive your org-specific data encryption key.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select a key type.
3. Click **Bring Your Own Key**.
4. In the Upload Tenant Secret section, attach both the encrypted key material and the hashed plaintext key material. Click **Upload**.



Bring Your Own Keys

Upload a tenant secret by first selecting or creating an active 4096-bit certificate in the Manage Certificates section. Then upload an encrypted secret and hashed tenant secret in the Upload Tenant Secret section.

« Back to Platform Encryption

Manage Certificates

Choose Certificate: my.cert

Certificate Details

Name	my.cert	Unique Name	my_cert
Created Date	8/18/2016 4:32 PM	Expiration Date	8/17/2018 5:00 PM
Key Size	4096		

Upload Tenant Secret

Encrypted Tenant Secret: No file chosen

Hashed Tenant Secret: No file chosen

This tenant secret automatically becomes the active tenant secret.

Your tenant secret is now ready to be used for key derivation. From here on, the Shield KMS uses your tenant secret to derive an org-specific data encryption key. The app server then uses this key to encrypt and decrypt your users' data.

If you don't want Salesforce to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. For more information, see Opt-Out of Key Derivation with BYOK in Salesforce Help.

Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Fields and Files (Probabilistic) tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you reach the limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data that it encrypts with an active key.

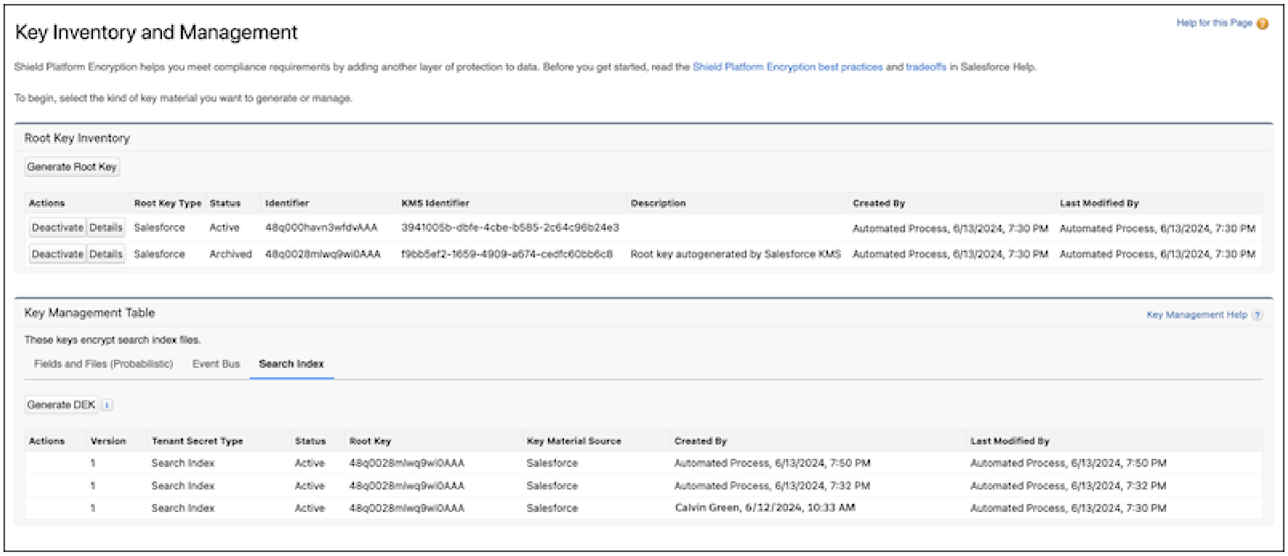
5. Export your tenant secret, and back it up as prescribed in your organization's security policy.

   To restore a destroyed tenant secret, reimport it. The exported tenant secret is different from the tenant secret you uploaded. It's encrypted with a different key and has additional metadata embedded in it. See Back Up Your Tenant Secret in Salesforce Help.

Upload Your BYOK DEK

After you have your BYOK-compatible DEK, upload it to Salesforce. The Shield Key Management Service (KMS) uses your DEK for encrypting and decrypting your search indexes. Currently a BYOK DEK is supported only for Search Index encryption. Before you can create a search index DEK, you must create a root key. It's the root key that creates the DEK and wraps it when necessary.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
   Salesforce shows the Key Inventory and Management page.



Key Inventory and Management

Shield Platform Encryption helps you meet compliance requirements by adding another layer of protection to data. Before you get started, read the Shield Platform Encryption best practices and tradeoffs in Salesforce Help.

To begin, select the kind of key material you want to generate or manage.

Root Key Inventory

Actions	Root Key Type	Status	Identifier	KMS Identifier	Description	Created By	Last Modified By
Deactivate Details	Salesforce	Active	48q000havn3wfdvAAA	3941005b-dbfe-4cbe-b585-2c64c96b24e3		Automated Process, 6/13/2024, 7:30 PM	Automated Process, 6/13/2024, 7:30 PM
Deactivate Details	Salesforce	Archived	48q0028mlwq9wi0AAA	f9bb5ef2-1659-4909-a674-cedfc60bb6c8	Root key autogenerated by Salesforce KMS	Automated Process, 6/13/2024, 7:30 PM	Automated Process, 6/13/2024, 7:30 PM

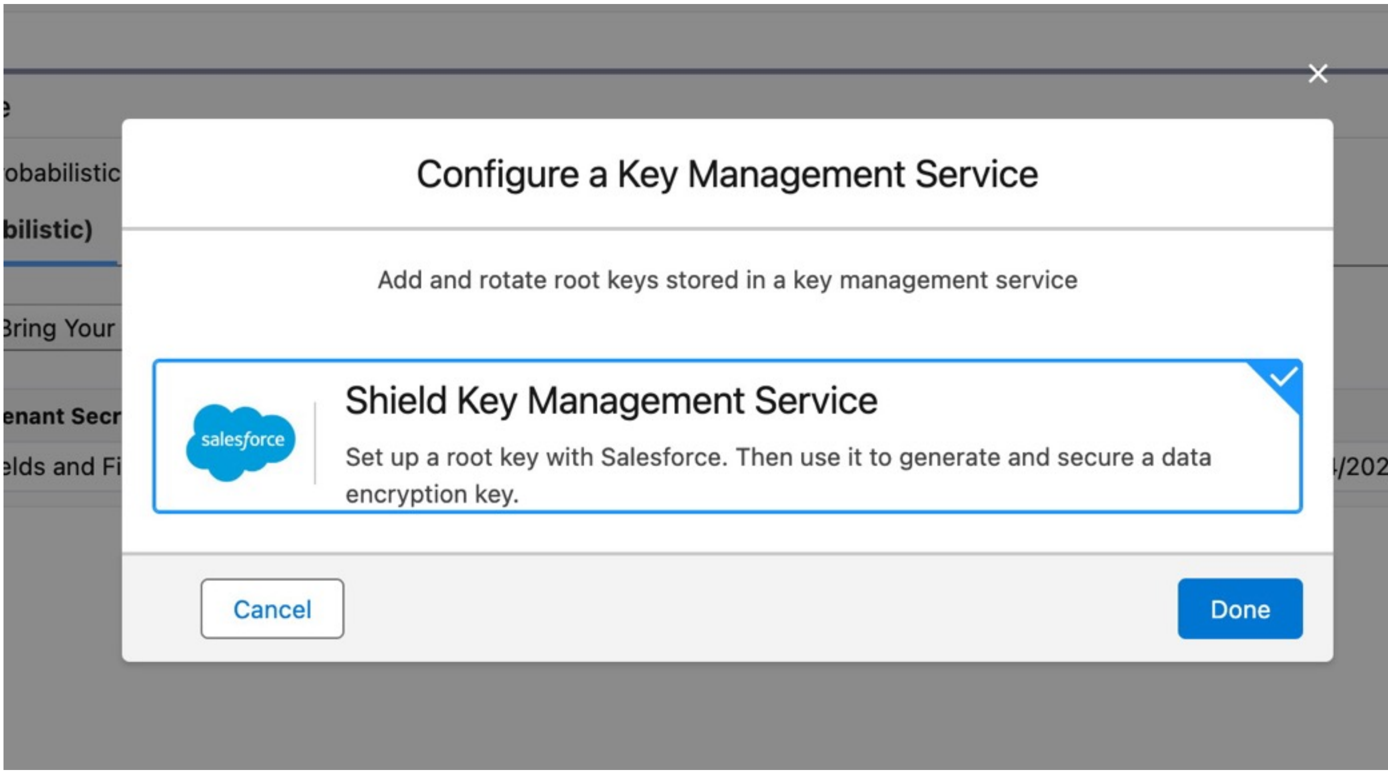
Key Management Table

These keys encrypt search index files.

Actions	Version	Tenant Secret Type	Status	Root Key	Key Material Source	Created By	Last Modified By
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Automated Process, 6/13/2024, 7:50 PM	Automated Process, 6/13/2024, 7:50 PM
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Automated Process, 6/13/2024, 7:32 PM	Automated Process, 6/13/2024, 7:32 PM
	1	Search Index	Active	48q0028mlwq9wi0AAA	Salesforce	Calvin Green, 6/12/2024, 10:33 AM	Automated Process, 6/13/2024, 7:30 PM

2. In the Root Key Inventory table, check that a root key exists. If a root key exists, go on to step 3.

   a. Click **Generate Root Key**.
      The Configure a Key Management Service dialog appears



Configure a Key Management Service

Add and rotate root keys stored in a key management service

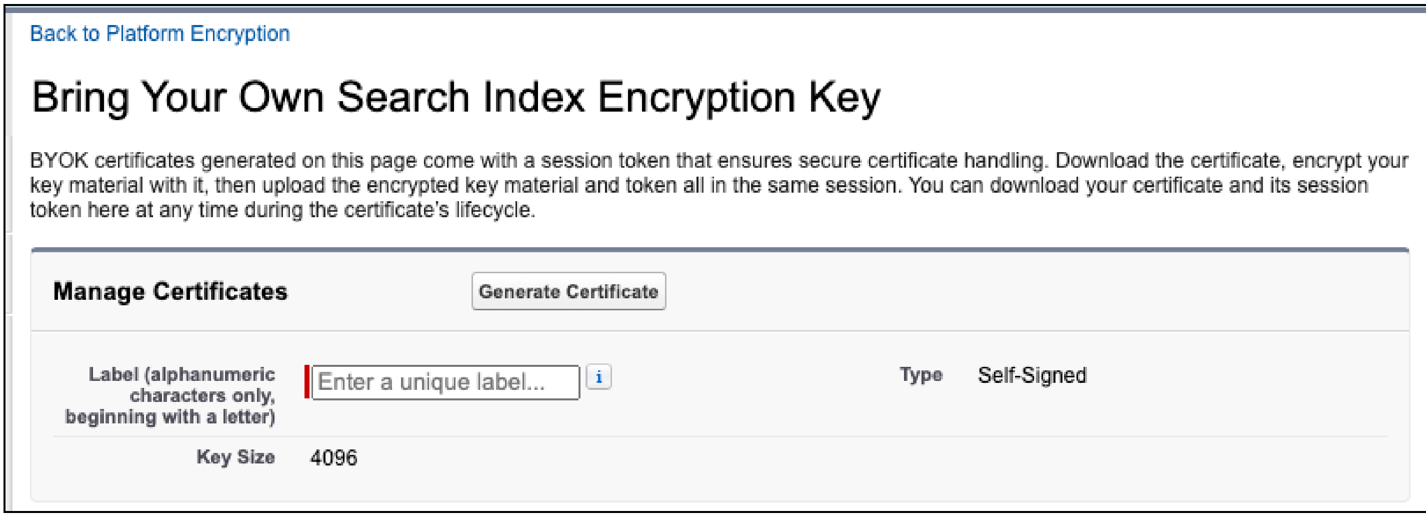
Shield Key Management Service

Set up a root key with Salesforce. Then use it to generate and secure a data encryption key.

.

b. Click **Shield Key Management Service** and then click **Done**.
   Salesforce begins the process for generating the root key. This can take a while. You're notified by email when the root key is ready. When you have confirmation, go on to the next step.

3. In the Key Management Table, select **Search Index**.
4. Click **Generate DEK**.
   Salesforce uses the root key to generate a DEK. This can take a while. You're notified by email when the root key is ready.
5. Click **Bring Your Own Key**.
   If you're prompted to generate a certificate, enter an alphanumeric label and then select **Generate Certificate.**



Back to Platform Encryption

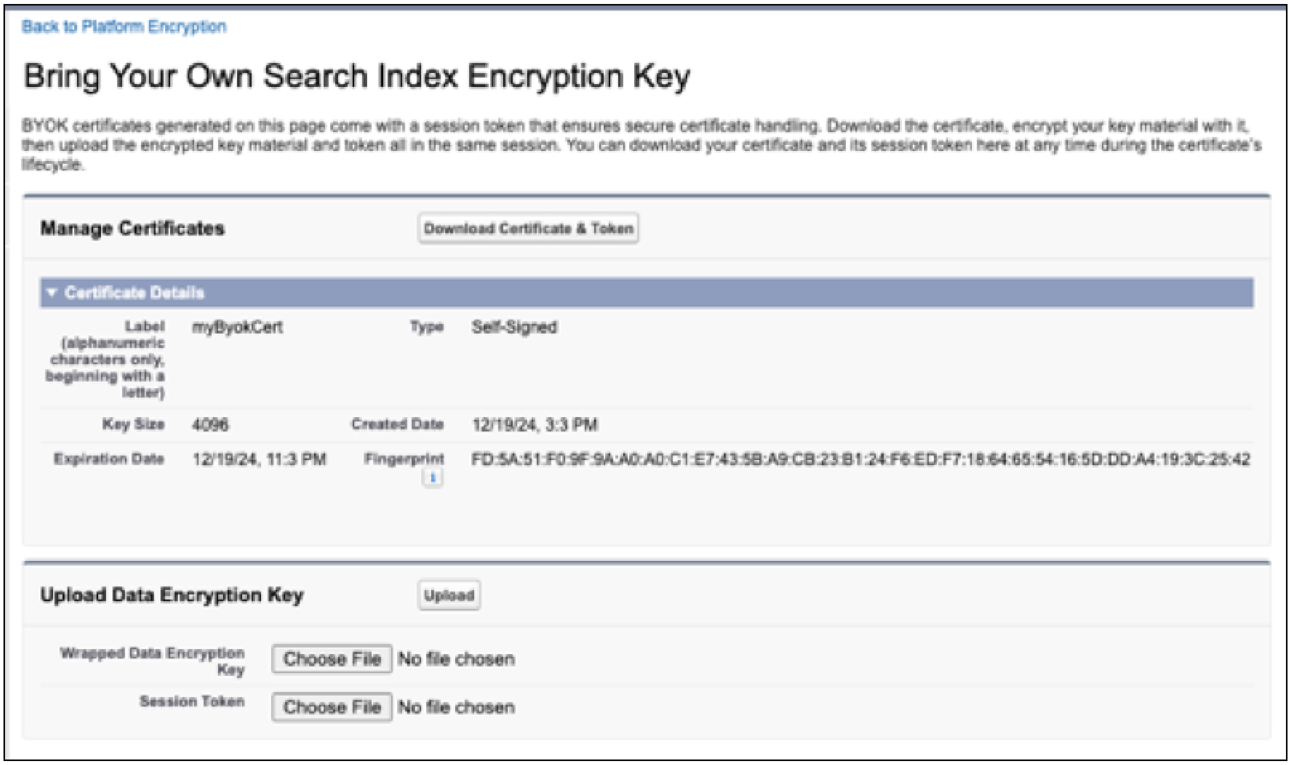
Bring Your Own Search Index Encryption Key

BYOK certificates generated on this page come with a session token that ensures secure certificate handling. Download the certificate, encrypt your key material with it, then upload the encrypted key material and token all in the same session. You can download your certificate and its session token here at any time during the certificate's lifecycle.

Manage Certificates

Label (alphanumeric characters only, beginning with a letter)	Enter a unique label...	Type	Self-Signed
Key Size	4096		

6. In the Upload Data Encryption Key section, attach both the encrypted key material and the hashed plaintext key material. Click **Upload**.



Back to Platform Encryption

Bring Your Own Search Index Encryption Key

BYOK certificates generated on this page come with a session token that ensures secure certificate handling. Download the certificate, encrypt your key material with it, then upload the encrypted key material and token all in the same session. You can download your certificate and its session token here at any time during the certificate's lifecycle.

Manage Certificates

▼ Certificate Details

Label (alphanumeric characters only, beginning with a letter)	myByokCert	Type	Self-Signed
Key Size	4096	Created Date	12/19/24, 3:3 PM
Expiration Date	12/19/24, 11:3 PM	Fingerprint	FD:5A:51:F0:9F:9A:A0:A0:C1:E7:43:5B:A9:CB:23:B1:24:F6:ED:F7:18:64:65:54:16:5D:DD:A4:19:3C:25:42

Upload Data Encryption Key

Wrapped Data Encryption Key	No file chosen
Session Token	No file chosen

This DEK automatically becomes the active data encryption key for Search Indexes.

From here on, the Shield KMS uses your DEK to encrypt and decrypt your users' search data.

Opt Out of Key Derivation with BYOK

If you don't want Shield Platform Encryption to derive a data encryption key for you, you can opt out of key derivation and upload your own DEK. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

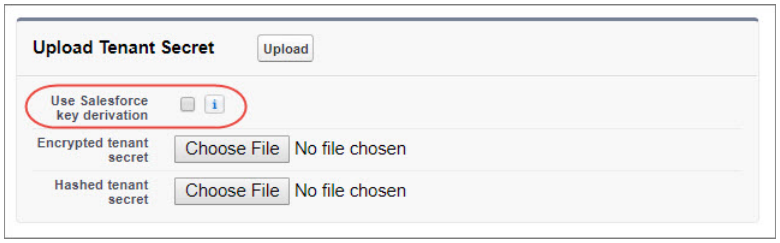
Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Generate your customer-supplied data encryption key using a method of your choice. Then calculate an SHA256 hash of the key, and encrypt it with the public key from a BYOK-compatible certificate. See Upload Your BYOK Key Material for details about how to prepare customer-supplied key material.

1. Make sure that your org has the Bring Your Own Keys feature enabled. To enable this feature, contact Salesforce Customer Support.
2. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
3. In the Advanced Encryption Settings section, turn on **Allow BYOK to Opt-Out of Key Derivation.**

   You can also enable the Allow BYOK to Opt-Out of Key Derivation setting programmatically. See EncryptionKeySettings in the *Metadata API Developer Guide*.

   You can now opt out of key derivation when you upload key material.
4. From Setup, in the Quick Find box, enter `Key Management`, and then select **Key Management**.
5. In the Key Management Table, select a key type.
6. Click **Bring Your Own Key**.
7. Deselect **Use Salesforce key derivation**.


Upload Tenant Secret
Upload
Use Salesforce key derivation □ i
Encrypted tenant secret
Choose File No file chosen
Hashed tenant secret
Choose File No file chosen

8. In the Upload Tenant Secret section, attach your encrypted data encryption key and your hashed plaintext data encryption key.
9. Click **Upload**.
   This data encryption key automatically becomes the active key. From now on, the Shield Key Management Service (KMS) skips the derivation process and uses your data encryption key to directly encrypt and decrypt your data. You can review the derivation status of all key material on the Key Management page.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

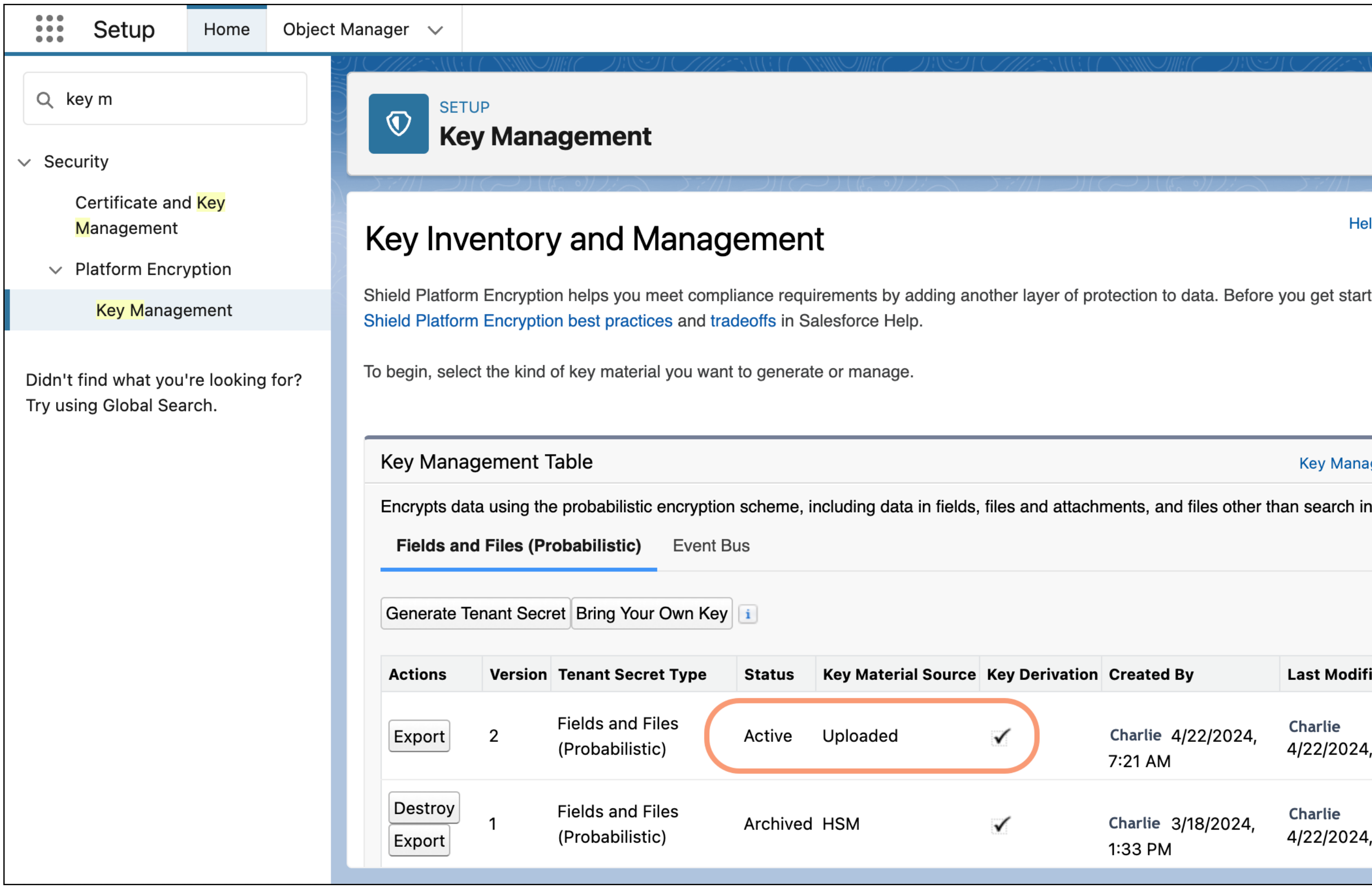
USER PERMISSIONS

To generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:
- Manage Encryption Keys

To allow BYOK to opt out of key derivation:
- Customize Application

  AND

  Manage Encryption Keys



Didn't find what you're looking for? Try using Global Search.

Key Inventory and Management

Shield Platform Encryption helps you meet compliance requirements by adding another layer of protection to data. Before you get start Shield Platform Encryption best practices and tradeoffs in Salesforce Help.

To begin, select the kind of key material you want to generate or manage.

Key Management Table

Encrypts data using the probabilistic encryption scheme, including data in fields, files and attachments, and files other than search in

Fields and Files (Probabilistic) | Event Bus

Actions	Version	Tenant Secret Type	Status	Key Material Source	Key Derivation	Created By	Last Modifi
Export	2	Fields and Files (Probabilistic)	Active	Uploaded	✓	Charlie 4/22/2024, 7:21 AM	Charlie 4/22/2024,
Destroy Export	1	Fields and Files (Probabilistic)	Archived	HSM	✓	Charlie 3/18/2024, 1:33 PM	Charlie 4/22/2024,

10. Export your data encryption key and back it up as prescribed in your organization's security policy.

    To restore your data encryption key, reimport it. The exported data encryption key is different from the data encryption key that you uploaded. It's encrypted with a different key and has additional embedded metadata. See Back Up Your Tenant Secret in *Salesforce Help*.

Take Good Care of Your BYOK Keys

When you create and store your own key material outside of Salesforce, it's important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

Back up all imported key material after you upload them to Salesforce. Backing it up ensures that you have copies of your active key material. See Back Up Your Tenant Secret in Salesforce Help.

Review your company policy on key rotation. You can rotate and update your keys on your own schedule. See Rotate Your Encryption Keys.

Important: If you accidentally destroy a tenant secret or DEK that isn't backed up, Salesforce can't help you retrieve it.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Troubleshooting Bring Your Own Key

Read these frequently asked questions to help you troubleshoot any problems that arise with Shield Platform Encryption's Bring Your Own Key service.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

I'm trying to use the script you provide, but it doesn't run.

Make sure that you're running the right script for your operating system. If you're working on a Windows machine, you can install a Linux emulator and use the Linux script. These issues can also prevent the script from running:

- You don't have write permission in the folder you're trying to run the script from. Try running the script from a folder that you have write permission for.
- The certificate that the script references is missing. Make sure you've properly generated the certificate.
- The certificate is missing or isn't being referenced by the correct name. Make sure you've entered the correct file name for your certificate in the script.

I want to use the script you provide, but I also want to use my own random number generator.

The script we provide uses a random number generator to create a random value that is then used as your tenant secret. If you want to use a different generator, replace `head -c 32 /dev/urandom | tr '\n' =` (or, in the Mac version, `head -c 32 /dev/urandom > $PLAINTEXT_SECRET`) with a command that generates a random number using your preferred generator.

What if I want to use my own hashing process to hash my tenant secret?

No problem. Make sure that the result meets these requirements:

- Uses an SHA-256 algorithm.
- Results in a base64 encoded hashed tenant secret.
- Generates the hash of the random number BEFORE encrypting it.

If any of these three criteria aren't met, you can't upload your tenant secret.

How should I encrypt my tenant secret before I upload it to Salesforce?

If you're using the script provided, the encryption process is taken care of. If you don't use the script, specify the OAEP padding scheme when you encrypt your tenant secret. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64. If either of these criteria aren't met, you can't upload your tenant secret.

If you choose to not use the script provided, follow the instructions in the Generate And Wrap Your Tenant Secret Help topic.

My wrapped DEK isn't accepted. What do I do?

Make sure that you wrap your root-key generated DEKs (such as for Search Index Encryption and Database Encryption) with the public key from the BYOK-compatible certificate that you generated by using the SHA512 padding algorithm. Wrap your other BYOK tenant secrets by using the SHA1 algorithm.

My certificate is about to expire. What do I do?

An expired certificate doesn't affect the active state of the secret that it wraps. Your certificate gives assurance to the recipient that the received secret was sent and wrapped by you. If you use an expired certificate, your secret is still protected, but the receiving party is notified that the certificate is expired. Salesforce doesn't block your secret if it's wrapped with an expired certificate. Note that you can't upload a new secret or DEK using an expired secret.

I can't upload my Encrypted tenant secret and Hashed tenant secret.

A handful of errors can prevent your files from uploading. Use the chart to make that sure your tenant secrets and certificates are in order.

Possible cause	Solution
Your files were generated with an expired certificate.	Check the date on your certificate. If it has expired, you can renew your certificate or use another one.
Your certificate isn't active, or isn't a valid Bring Your Own Key certificate.	Ensure that your certificate settings are compatible with the Bring Your Own Key feature. Under the Certificate and Key Edit section of the Certificates page, select a 4096-bit certificate size, disable Exportable Private Key, and enable Platform Encryption. Read more about expired certificates in the "My certificate is about to expire" section.
You haven't attached both the encrypted tenant secret and the hashed tenant secret.	Make sure that you attach both the encrypted tenant secret and the hashed tenant secret. Both of these files should have a .b64 suffix.
Your tenant secret or hashed tenant secret wasn't generated properly.	Several problems can cause this error. Usually, the tenant secret or hashed tenant secret wasn't generated using the correct SSL parameters. If you're using OpenSSL, you can refer to the script for an example of the correct parameters you should use to generate and hash your tenant secret. If you're using a library other than OpenSSL, check that library's support page for help with finding the correct parameters to both generate and hash your tenant secret. Still stuck? Contact your Salesforce account executive. They'll put you in touch with someone at Salesforce who can help.

I'm still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive. They'll put you in touch with a support team specific to this feature.

External Key Management

Shield External Key Management (EKM) connects your Salesforce implementation to your keys in AWS KMS and uses those keys for encryption operations on Salesforce data. EKM fetches your keys on demand from AWS KMS over a secure channel. EKM stores your key in the key cache and uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cached EKM keys in any system of record or backups. You can revoke key material at any time.

When you encrypt data using EKM, you get the benefits built into Salesforce Shield Platform Encryption plus the extra assurance that comes from managing your keys with your preferred key management service. Unlike Salesforce's Cache-Only Key Service, EKM integrates natively with external key management services for a quicker, more streamlined user experience.

Note: Salesforce EKM currently supports AWS Key Management Service key material only. Refer to the AWS KMS documentation for information about creating, accessing, and managing keys in AWS.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

How Salesforce Shield EKM Works

For EKM, Shield Platform Encryption relies on the customer's external KMS to generate and secure the data encryption keys (DEKs) used by the Shield Platform encryption service. These DEKs reside with the Shield Platform encrypted key cache in a wrapped state. When encryption or decryption operations are needed, the Shield Platform service passes the wrapped DEK to the customer's external key service to be unwrapped. The customer key service unwraps the DEK and sends it securely back to the Shield Platform encryption service.

EKM Prerequisites

To use EKM, you must create a data encryption key (DEK) of sufficient strength in a supported external key management service. You should also check that an external application can communicate with the key service to securely retrieve the DEK.

Key Coordination Policy Setup

Track the status of both the AWS key and the Salesforce EKM key that depends on it.

EKM Considerations

Take care when managing your external keys. Your Salesforce application depends on your external keys to encrypt and decrypt your data. If the key status changes, your users could permanently lose access to encrypted data.

Connect Salesforce to AWS KMS and Create a Data Encryption Key

When you configure your connection between Salesforce and AWS, you provide information about the AWS KMS key that you want Salesforce to use (key identifier, region, and description). You then generate a JSON structure and add that structure to your key policy in the AWS console for your key.

Key Maintenance and Auditing for EKM

Common key operations include auditing, deactivating, reactivating, rotating, and checking the connection to your external keys. These operations affect the keys identified in your Salesforce setup. The original keys in AWS are managed by a separate AWS process.

EKM in a Sandbox Org

A sandbox org that's copied, refreshed, or cloned from a source org that uses EKM keys is granted minimum access to the source org's keys, so that it can decrypt any encrypted data it inherited from the source org. A sandbox org can't manage its source org's keys in any way, because sandboxes have limited access to those keys. Rotate the keys in a sandbox org as soon as you create it.

How Salesforce Shield EKM Works

For EKM, Shield Platform Encryption relies on the customer's external KMS to generate and secure the data encryption keys (DEKs) used by the Shield Platform encryption service. These DEKs reside with the Shield Platform encrypted key cache in a wrapped state. When encryption or decryption operations are needed, the Shield Platform service passes the wrapped DEK to the customer's external key service to be unwrapped. The customer key service unwraps the DEK and sends it securely back to the Shield Platform encryption service.

The process begins when you create a root key in the customer KMS. You create a policy which gives Salesforce's regional KMS some important permissions.

- Permission to request the customer key service to generate and wrap a DEK by using the root key
- Permission to request the customer key service to unwrap the DEK by using the customer root key

You use this policy to create an EKM DEK in Setup. Then the Shield Platform encryption service requests the customer KMS to generate a DEK by using the root key. The customer KMS creates a DEK, wraps it, and sends it to the Shield Platform encryption service over a secure channel. This is the only copy of the DEK that exists. Shield Platform Encryption stores the DEK, still wrapped by the root key, in the TenantSecret database. Here's the process, step by step:

1. The customer KMS admin creates a root key.
2. The Salesforce admin creates a key policy and copies it to the customer KMS.
3. With the policy in place, the Salesforce encryption service requests a DEK for local storage.
4. The customer KMS uses the root key to create and wrap the new DEK, which it sends back via a secure channel.
5. The encryption service stores the wrapped DEK in the TenantSecret table.

EDITIONS

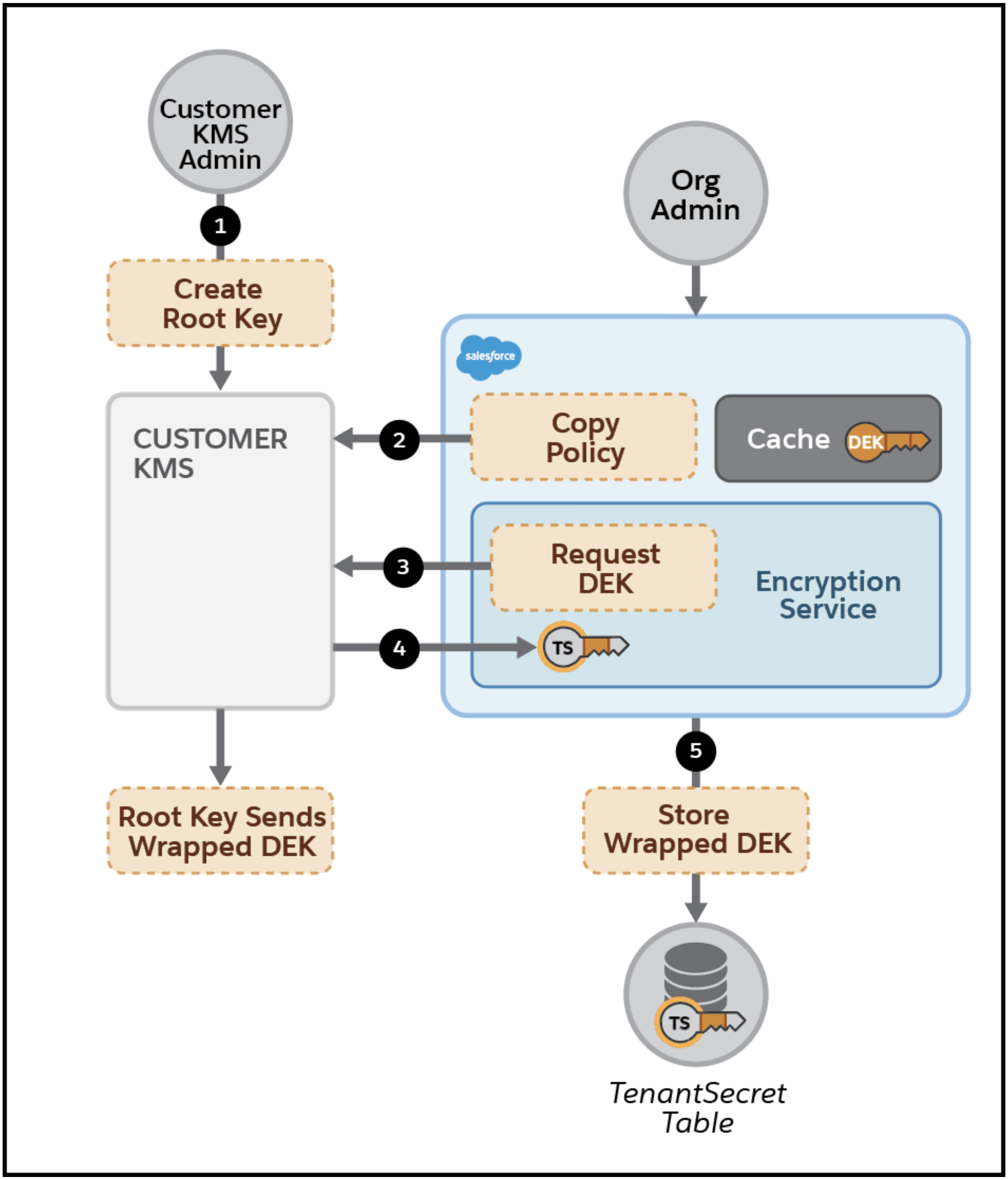
Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

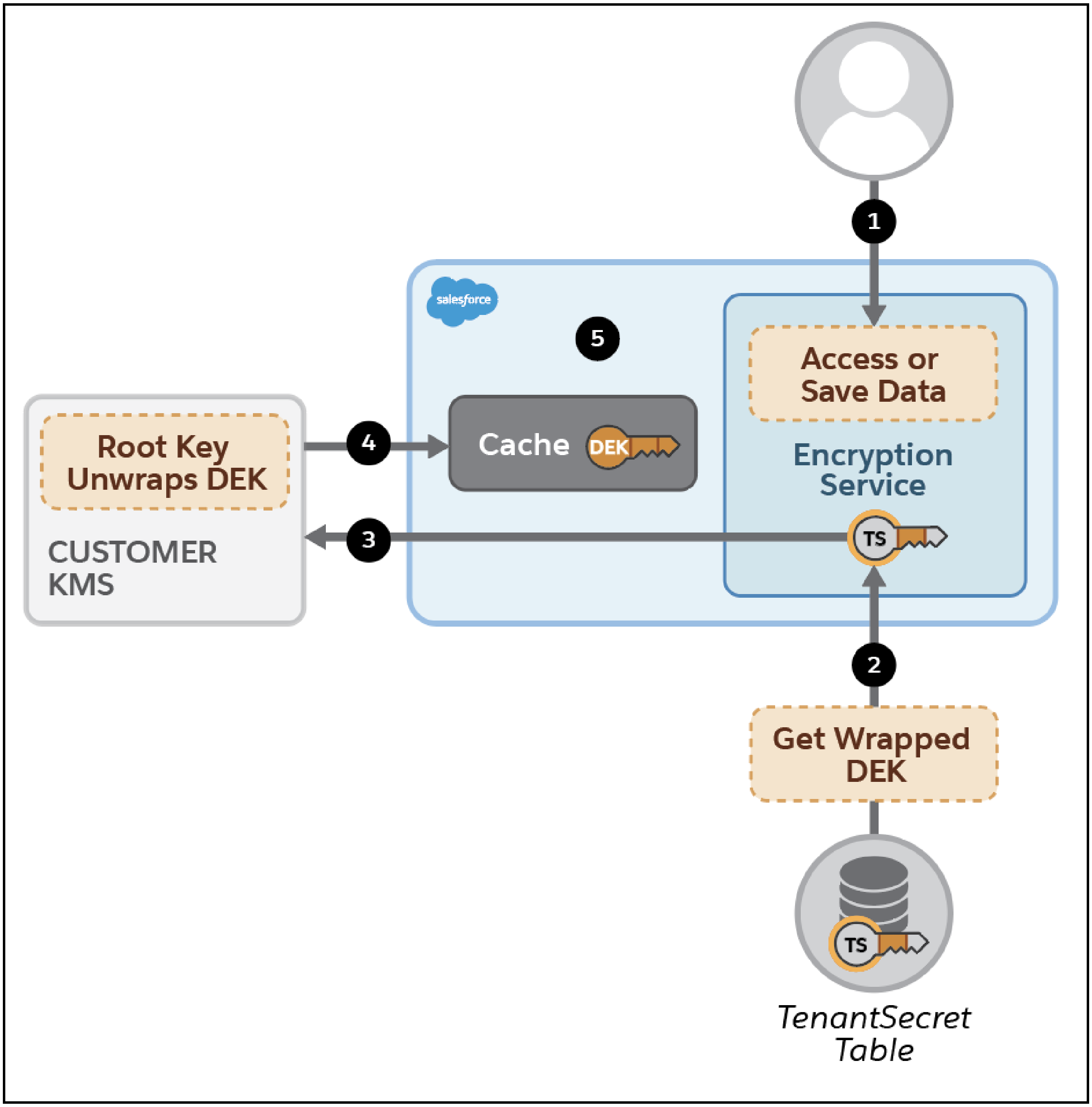
- Manage Encryption Keys



When the Shield Platform encryption service detects encryption operations that require the EKM DEK, it checks its encrypted key cache for it. If the unwrapped DEK isn't present in the cache, the Shield Platform encryption service requests that the key service unwrap the

DEK. The key service unwraps the DEK and sends it back to the Shield Platform encryption service over a secure channel (TLS(Awskms-SFKMS)/mTls). Then the Shield Platform encryption service adds the unwrapped key to the encrypted key cache.

1. A user accesses or saves encrypted data.
2. The Shield Platform encryption service gets the DEK from the TenantSecret table.
3. The encryption service sends the wrapped key to the customer KMS over a secure channel to be unwrapped.
4. The customer KMS uses the root key to unwrap the DEK and sends it back to the encryption service.
5. The encryption service stores the unwrapped key in the encrypted key cache for immediate use.



If the unwrapped DEK is present in the cache, the Shield Platform encryption service uses it for encryption and decryption of customer data.

Because EKM DEKs bypass the key-derivation process, they're used to directly encrypt and decrypt your data.

As a core offering of the Shield KMS, enhanced cache controls ensure that key material is stored securely while in the cache. The Shield KMS encrypts the fetched key material with an org-specific AES 256-bit cache encryption key and stores the encrypted key material in the cache for encrypt and decrypt operations. HSM-protected keys secure the cache encryption key in the cache, and the cache encryption key is rotated along with key lifecycle events such as key destruction and rotation.

The enhanced cache controls provide a single source of truth for key material that's used to encrypt and decrypt your data. Subsequent encryption and decryption requests go through the encrypted key cache. They are unwrapped by the customer KMS until the DEK is revoked or rotated or when the cache is flushed. After the cache is flushed, the EKM service again fetches the DEK from your specified key service. The cache is flushed regularly every 72 hours. Certain Salesforce operations flush the cache, on average, every 24 hours. Destroying a DEK invalidates the corresponding DEK that's stored in the cache.

EKM Prerequisites

To use EKM, you must create a data encryption key (DEK) of sufficient strength in a supported external key management service. You should also check that an external application can communicate with the key service to securely retrieve the DEK.

Salesforce EKM supports AWS Key Management Service key material only. Refer to the AWS KMS documentation for information about creating, accessing, and managing keys in AWS.

Before you configure your connection in Salesforce, create your key material in AWS KMS. Salesforce requires:

- Symmetric key type
- Single region (MultiRegion = False)
- An ARN that's in the same AWS region as the current Hyperforce instance within which your core org resides.

Make sure that you can access key material in both Salesforce and AWS KMS.

Exercise careful accounting between the Salesforce Key Management Setup page and the AWS KMS dashboard. AWS KMS has no information about the status of Salesforce EKM secrets.

SEE ALSO:

Check the Connection to Your EKM Key

Key Coordination Policy Setup

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Key Coordination Policy Setup

Track the status of both the AWS key and the Salesforce EKM key that depends on it.

The relationship between the AWS KMS key and the Salesforce EKM key is one way. Though the EKM key refers directly to the AWS key, the AWS key has no reference back to the EKM key. If the AWS key is inadvertently deleted, encryption and decryption continue until the AWS key is flushed from the cache. After the AWS key is flushed from the cache, no decryption of data that was encrypted with the matching EKM key is possible.

Set up an operational accounting policy that governs how the key states are communicated and managed. If you no longer need an EKM key, you can deactivate it on the Key Management page in Setup. But what do you do with the AWS key? We recommend that you back it up. To avoid losing access to data, document the who, what, when, where, why, and how of all your key relationships. Make that documentation available to the people who need it.

SEE ALSO:

Set Up Your Encryption Policy

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

EKM Considerations

Take care when managing your external keys. Your Salesforce application depends on your external keys to encrypt and decrypt your data. If the key status changes, your users could permanently lose access to encrypted data.

- Make sure that your encryption policy includes key-rotation and key-backup strategies as safeguards against unplanned key loss. Deactivate and destroy operations evict encrypted key material from the cache. If the external key or the associated Salesforce data encryption keys are disabled, deactivated, or deleted, related Salesforce data encrypted with them is no longer accessible.
- External keys created in production can't be activated or deactivated in sandboxes. As a best practice, rotate data encryption keys in sandboxes immediately after a refresh. Rotation ensures that production and sandbox orgs use different data encryption keys, and that you'll have full control over them.
- If a key isn't available on the AWS side, after the key is flushed from the cache, neither encryption nor decryption is possible. Users who try to access encrypted data see three question marks (`???`) instead of the ciphertext. Any attempts to write data to encrypted fields fail. Users see an error message that says the key is unavailable.
- When the AWS key isn't available, we change the status of the key to `Unavailable`. This means we stop trying to call AWS KMS to get the key. You can check the connection to attempt to reconnect to the key and update its status.
- If you're using EKM, you can still rotate the other types of keys available to your product (EKM, BYOK, Cache-only key, or a Salesforce-generated key).

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

SEE ALSO:

How Shield Platform Encryption Works in a Sandbox

Set Up Your Encryption Policy

Check the Connection to Your EKM Key

Connect Salesforce to AWS KMS and Create a Data Encryption Key

EKM Prerequisites

Connect Salesforce to AWS KMS and Create a Data Encryption Key

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

When you configure your connection between Salesforce and AWS, you provide information about the AWS KMS key that you want Salesforce to use (key identifier, region, and description). You then generate a JSON structure and add that structure to your key policy in the AWS console for your key.

> Important: Before you can use EKM, you must create and configure the AWS key you plan to use. See the AWS Key Management Service documentation.

You can also add information about your Salesforce key policy to your key policy in AWS KMS. Salesforce then uses this key policy to generate and wrap a data encryption key for encryption and decryption operations in Salesforce.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Advanced Settings**. Turn on **External Key Management**.
   You can now access External Key Management configuration controls on the Key Management page.
2. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
3. Click **Manage External Keys**.
4. Select **AWS Key Management Service**, and then click **Start**.
5. Follow the prompts for gathering and entering your AWS KMS key information. Enter its key identifier, region, and description. A unique description helps you distinguish between keys for efficient auditing and key management.
6. To create a copy of the JSON text, on the Key Policy tab, click **Copy**.

   The copied JSON text contains details about your AWS KMS key that you entered in the previous step.
7. Log in to your AWS KMS console. Paste the copied JSON text into your key policy. Make sure that it references your key ID and not an alias name, and then save your changes.
   For example, use `key/`***`key_id`*** instead of `alias/`***`alias_name`*** in your ARN.
8. In Salesforce, on the Key Management page, click **Done**.

You receive a notification that AWS KMS is now connected to Salesforce and that a Salesforce data encryption key is created. Check the connection and new data encryption key on the Key Management page.

SEE ALSO:

Check the Connection to Your EKM Key

Key Maintenance and Auditing for EKM

Common key operations include auditing, deactivating, reactivating, rotating, and checking the connection to your external keys. These operations affect the keys identified in your Salesforce setup. The original keys in AWS are managed by a separate AWS process.

Audit an EKM Key

In this context, auditing means examining the details about the EKM key, such as when it was last modified. You can also view each external key's unique policy.

Deactivate an EKM Key

When you want to revoke all access to encrypted data, or rotate keys as a part of planned maintenance, you can deactivate key material. The effect of deactivating key material is similar to that of deleting a key. Your data remains encrypted, but it can't be decrypted.

Reactivate an EKM Key

You can make a previously deactivated key active again. When a key is reactivated, data previously encrypted with the key can be decrypted and viewed.

Rotate an EKM Key

Key rotation refers to the process of updating or changing your key material. You can edit existing key materials or replace them with new ones. If you edit or update your external key, make sure to align your external key details across both Salesforce and AWS KMS.

Check the Connection to Your EKM Key

You can check the connection between Salesforce and your external key management service. This information can help you troubleshoot problems when you configure your key policy.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Audit an EKM Key

In this context, auditing means examining the details about the EKM key, such as when it was last modified. You can also view each external key's unique policy.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the External Key Inventory, click **Details**.

For a list of past actions taken on the key management page, visit Setup Audit Trail.

SEE ALSO:

Monitor Setup Changes with Setup Audit Trail

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure Shield Platform Encryption key material:

- Manage Encryption Keys

Deactivate an EKM Key

When you want to revoke all access to encrypted data, or rotate keys as a part of planned maintenance, you can deactivate key material. The effect of deactivating key material is similar to that of deleting a key. Your data remains encrypted, but it can't be decrypted.

Consider the effect on your users and data of deactivating the EKM key. Data encrypted with the key isn't decryptable. Make sure that the data you need is synchronized to a different key.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the External Key Inventory, click **Details** for the key you want to deactivate.
3. In the pane that opens, review the information. Then click either **Never Mind** or **Deactivate External Key**.

Communicate with any other key managers that the key is now deactivated. Be alert for users reporting an inability to access encrypted data they could see previously.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Reactivate an EKM Key

You can make a previously deactivated key active again. When a key is reactivated, data previously encrypted with the key can be decrypted and viewed.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the External Key Inventory, click **Activate** next to the key you want to activate.

Check that you can view data previously encrypted using the reactivated key. Communicate with any other key managers that the key is now reactivated.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Rotate an EKM Key

Key rotation refers to the process of updating or changing your key material. You can edit existing key materials or replace them with new ones. If you edit or update your external key, make sure to align your external key details across both Salesforce and AWS KMS.

Keep these considerations in mind when rotating external keys.

- If you deactivate or destroy external keys, encrypted key material is evicted from the cache.
- If you disable, deactivate, or delete the external key or an associated Salesforce data-encryption key, related Salesforce data encrypted with that key is no longer accessible.
- As a best practice, rotate data encryption keys in sandboxes after a refresh. Rotation ensures that production and sandbox orgs use different data encryption keys. You can't activate or deactivate in a sandbox an external key created in production.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Click **Manage External Keys**.
3. Choose to either use the latest configuration of the current key or to use a different key.
4. Complete the steps on screen.

Store or version your old keys securely, in case you need them again someday. Communicate the change you made so others who need to know are aware.

SEE ALSO:

Key Management and Rotation

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Check the Connection to Your EKM Key

You can check the connection between Salesforce and your external key management service. This information can help you troubleshoot problems when you configure your key policy.

Before you can check a key connection, you must set up a key policy on page 34.

Check the connection anytime you want to verify an accessible connection.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the External Key Inventory table, click **Details**.
3. In the KMS Connection Status section, click **Check**.
   You see details about your connection, such as whether the connection is successful and the unique key identifier used. If the connection is unsuccessful, you see an error that explains what went wrong. Use the information in this error to correct the issue.
4. If a key is listed as `Unavailable`, click **Retry**.
   This calls out to AWS to check whether the key works now and, if so, update the state.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

EKM in a Sandbox Org

A sandbox org that's copied, refreshed, or cloned from a source org that uses EKM keys is granted minimum access to the source org's keys, so that it can decrypt any encrypted data it inherited from the source org. A sandbox org can't manage its source org's keys in any way, because sandboxes have limited access to those keys. Rotate the keys in a sandbox org as soon as you create it.

When you create, refresh or clone a sandbox, the sandbox retains limited (read only) access to keys that were used to encrypt data the sandbox inherits. This is so you can decrypt the content.

Providing limited EKM key access is essential to ensure a consistent experience in your sandbox orgs. We strongly recommend that you rotate your keys on newly created sandbox orgs and sync your data via Encryption Statistics right away. By rotating your keys, you avoid complications that could happen if the original encryption keys are deactivated or destroyed. More specifically:

- In order to access their source org's keys, sandboxes must share their source org's region when using EKM.
- Consider changes in the source org's AWS KMS Key Policy that restrict source org access to data encryption keys. These changes propagate to the sandbox orgs that still depend on those keys at the time of change. If you rotate your keys, your sandbox is unaffected by changes in the source org's key policies.
- We recommend that you clone a sandbox only after you rotate your keys and sync all the encrypted data in the original sandbox.
- Access to keys is automatically extended at the time of sandbox creation, refresh or clone. We also remove such access to EKM-based keys at the time of permanent sandbox org deletion.
- When you clone a sandbox org (with EKM keys), access is extended only for the EKM keys that belong to the source sandbox org, not any keys that the sandbox org inherited between the time the original sandbox was created and the time the clone was created.

EDITIONS

Available in both Lightning Experience and Salesforce Classic (not available in all orgs).

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and either the EKM Service or the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

SEE ALSO:

Get Statistics About Your Encryption Coverage

Cache-Only Key Service

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce in any key repository or service that you control and have the Cache-Only Key Service fetch your key on demand from that key service. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

Note: Both BYOK and the Cache-Only Key service give you full control over which key service you use for your external keys. EKM supports only AWS KMS.

How Cache-Only Keys Works

The Shield Platform Encryption Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and Cache-Only Keys.

Prerequisites and Terminology for Cache-Only Keys

Shield Platform Encryption's Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key-management tasks. Before you start using the service, review how to create and host your key material in a way that's compatible with Salesforce's BYOK service. Also review several important terms relevant to the Cache-Only Key Service

Optimize Security Using Named Credentials and Cache-Only Keys

You can use an externally managed key as your cache-only key. External credentials create a secure connection between Salesforce and your external-key repository. For optimal security, set up an external credential that uses a named principal to authenticate into your external service on behalf of all users authorized to manage key material. Salesforce recommends you use this method instead of a legacy named credential if you use an external key management service along with cache-only keys.

Create and Assemble Your Key Material

The Shield Platform Encryption Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

Add Replay Detection for Cache-Only Keys

Replay detection protects your cache-only keys if a callout is fraudulently intercepted. When enabled, replay detection inserts an autogenerated, unique marker called a RequestIdentifier into every callout. The RequestIdentifier includes the key identifier, a nonce generated for that callout instance, and the nonce required from the endpoint. The RequestIdentifier serves as a random, one-time identifier for each valid callout request. After you set up your key service to accept and return the RequestIdentifier, any callout with missing or mismatched RequestIdentifiers is aborted.

Check Your Cache-Only Key Connection

Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

Destroy a Cache-Only Key

When you destroy a cache-only key, you're destroying two things: the key in the cache and the callout connection to the key service.

Reactivate a Cache-Only Key

If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup or programmatically through the API. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

Considerations for Cache-Only Keys

These considerations apply to all data that you encrypt using the Shield Platform Encryption Cache-Only Key Service.

Troubleshoot Cache-Only Keys

One or more of these frequently asked questions can help you troubleshoot any problems that arise with Shield Platform Encryption's Cache-Only Key Service.

SEE ALSO:

How Key Material Is Stored

External Key Management

How Cache-Only Keys Works

The Shield Platform Encryption Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

Figures 1 and 2 show how Salesforce fetches keys on-demand from your specified key service. Whether you store your keys with an on-premises key service or a cloud-based key service, the flow is the same. When users access encrypted data, or add sensitive data to encrypted data elements, the Cache-Only Key Service makes a callout to your key service. Your key service passes key material, wrapped securely in JSON Web Encryption format, through a secure, authenticated channel that you set up.

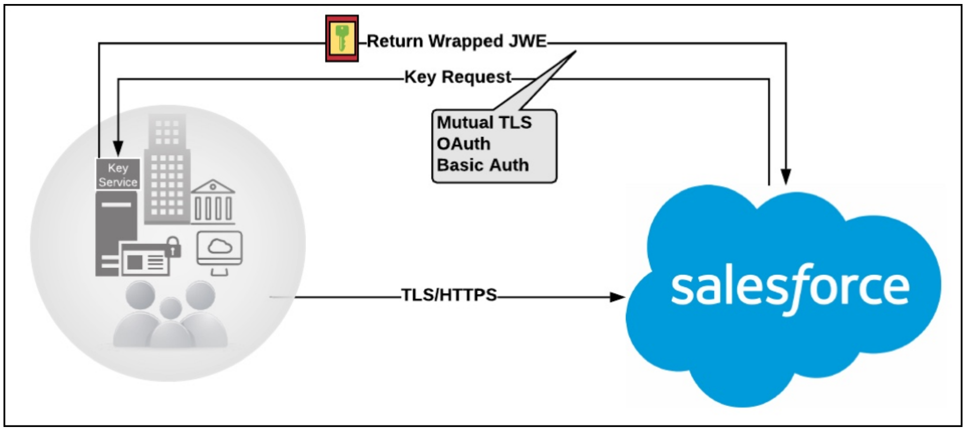


Figure 1: On-premises Key Service

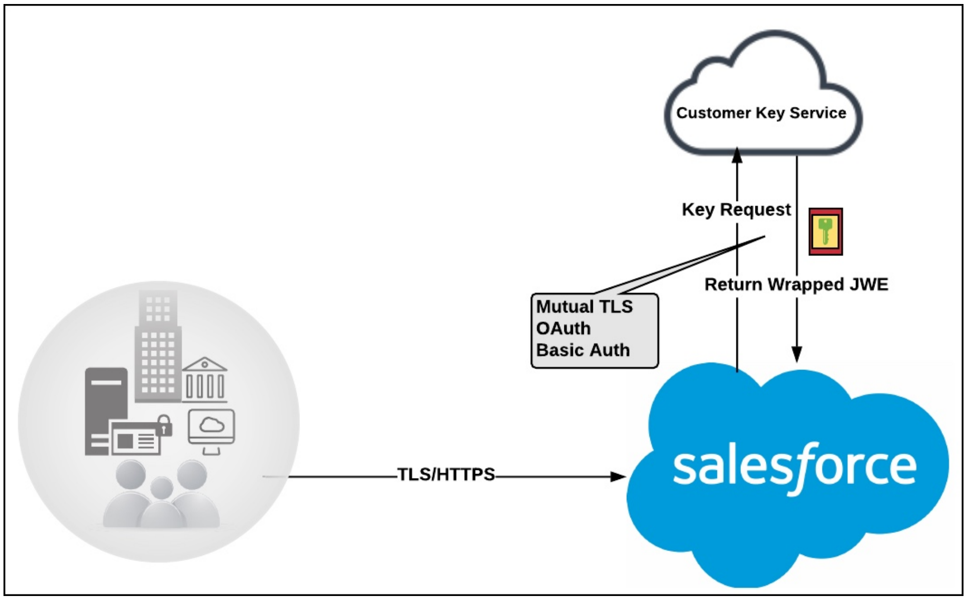


Figure 2: Cloud-Based Key Service

As a core offering of the Shield KMS, enhanced cache controls ensure that key material is stored securely while in the cache. The Shield KMS encrypts the fetched key material with an org-specific AES 256-bit cache encryption key and stores the encrypted key material in the cache for encrypt and decrypt operations. HSM-protected keys secure the cache encryption key in the cache, and the cache encryption key is rotated along with key lifecycle events such as key destruction and rotation.

The enhanced cache controls provide a single source of truth for key material used to encrypt and decrypt your data. Subsequent encryption and decryption requests go through the encrypted key cache until the cache-only key is revoked or rotated, or the cache is flushed. After the cache is flushed, the Cache-Only Key Service fetches key material from your specified key service. Shield Platform Encryption supports both named principals and legacy named credentials with no named principal. The cache is regularly flushed every 72 hours, and certain Salesforce operations flush the cache on average every 24 hours. Destroying a data encryption key invalidates the corresponding data encryption key that's stored in the cache.

Because cache-only keys bypass the key derivation process, they're used to directly encrypt and decrypt your data.

Prerequisites and Terminology for Cache-Only Keys

Shield Platform Encryption's Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key-management tasks. Before you start using the service, review how to create and host your key material in a way that's compatible with Salesforce's BYOK service. Also review several important terms relevant to the Cache-Only Key Service

Prerequisites

- The Cache-Only Key Service is available for tenant secrets only. It isn't compatible with root keys, such as those used with Search Index Encryption.
- Prepare your Salesforce org. Make sure that your org has at least one active Data in Salesforce key, either Salesforce-generated or one that you supply. You can create a tenant secret by clicking **Generate Tenant Secret** on the Key Management page in Setup.
- Generate and host key material. The cache-only key exchange protocol and format requires that keys are wrapped in an opinionated JSON Web Encryption (JWE). This format uses RSAES-OAEP for key encryption and AES GCM for content encryption.
- Use a secure, trusted service to generate, store, and back up your key material.
- Use and maintain a reliable high-availability key service. To mitigate any potential impact to business continuity, choose a high-availability key service with an acceptable service level agreement (SLA), predefined maintenance procedures, and processes.
- When the connection between Salesforce and your key service is broken, the Cache-Only Key Service can encrypt and decrypt data as long as your key material is in the cache. However, keys don't stay in the cache for long. The cache is regularly flushed every 72 hours, but some Salesforce operations flush the cache about every 24 hours.
- If your key material isn't in the cache and the connection to your key service is broken, users can't encrypt or decrypt records. Make sure that you use a key service that Salesforce can connect to at any time, especially during busy times, such as the end of the year or quarter.
- Maintain a secure callout endpoint. The cache-only key exchange protocol requires that keys are wrapped in an opinionated JSON format. Host your wrapped key inside the key response at a location Salesforce can request.
- The Cache-Only Key Service uses named credentials to establish a secure, authenticated connection to allowed IP addresses and domains. You can configure your named credentials to use popular authentication formats, such as Mutual TLS and OAuth. You can change these authentication protocols at any time.

  Note: A named credential for cache-only keys must specify a named principal. Creating a cache-only keys named credential requires the basic Named Credentials process with the added step of adding the `autoproc` user to a permission set. See Use a Named Principal-Based Credential for a Cache-Only Key for full details.

- Actively monitor your key service logs for errors. While Salesforce is here to help you with the Shield Platform Encryption service, you're responsible for maintaining the high-availability key service that you use to host your key material. You can use the RemoteKeyCalloutEvent object to review or track cache-only key events.

  Warning: Because you're in control of your keys, you're responsible for securing and backing up your key material. Salesforce can't retrieve lost key material stored outside of our encrypted key cache.

- Know how to format and assemble your key material. Format key material hosted outside of Salesforce in a way that's compatible with the Cache-Only Key Service. Make sure that you can generate these components in the required formats.

Table 3: Cache-Only Key Components

Component	Format
Data encryption key (DEK)	AES 256-bit
Content encryption key (CEK)	AES 256-bit

Component	Format
BYOK-compatible certificate	A 4096-bit RSA certificate whose private key is encrypted with a derived, org-specific tenant secret key
JSON Web Encryption content and header	See a sample in Github.
Algorithm for encrypting the CEK	RSA-OAEP
Algorithm for encrypting the DEK	A256GCM
Unique key identifier	Allows numbers, uppercase and lowercase letters, periods, hyphens, and underscores
Initialization vector	Encoded in base64url
JSON web token ID (JTI)	A 128-bit hex encoded, randomly generated identifier

Read more about assembling your key material in Create and Assemble Your Key Material on page 104. See Cache-Only Key Wrapper in GitHub for examples and a sample utility.

Terminology

Here are some terms that are specific to the Cache-Only Key Service.

Content Encryption Key
For each key request, your key service endpoint generates a unique content encryption key. The content encryption key wraps the data encryption key, which is then encrypted by the key encrypting key. After that it's placed in the JWE header of the key response.

JSON Web Encryption
The JSON-based structure that the Shield Platform Encryption service uses to encrypt content. JSON Web Encryption, or JWE, uses RSAES-OAEP for key encryption and AES GCM for content encryption.

JSON Web Token ID
A unique identifier for the JSON web token, which enables identity and security information to be shared across security domains.

Key Identifier
The Key ID (KID) is the unique identifier for your key. The KID is used as the suffix in the named credential and for validation of the KID in the response. In Setup, enter this identifier in the Unique Key Identifier field.

Optimize Security Using Named Credentials and Cache-Only Keys

You can use an externally managed key as your cache-only key. External credentials create a secure connection between Salesforce and your external-key repository. For optimal security, set up an external credential that uses a named principal to authenticate into your external service on behalf of all users authorized to manage key material. Salesforce recommends you use this method instead of a legacy named credential if you use an external key management service along with cache-only keys.

Before you begin, make sure to check the Prerequisites and Terminology for Cache-Only Keys. When you use a credential based on a named principal with your cache-only key, you provide both the location and the unique identifier for your key, so have those values ready before you begin.

To complete this process you will need the location URL and the unique ID of the external key. Please create your key material in your external KMS, and obtain the URL and ID before proceeding.

See Named Credentials.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and Cache-Only Keys.

USER PERMISSIONS

To create, edit, and delete named credentials:

- Customize Application

To allow cache-only keys with BYOK:

- Customize Application

  AND

  Manage Encryption Keys

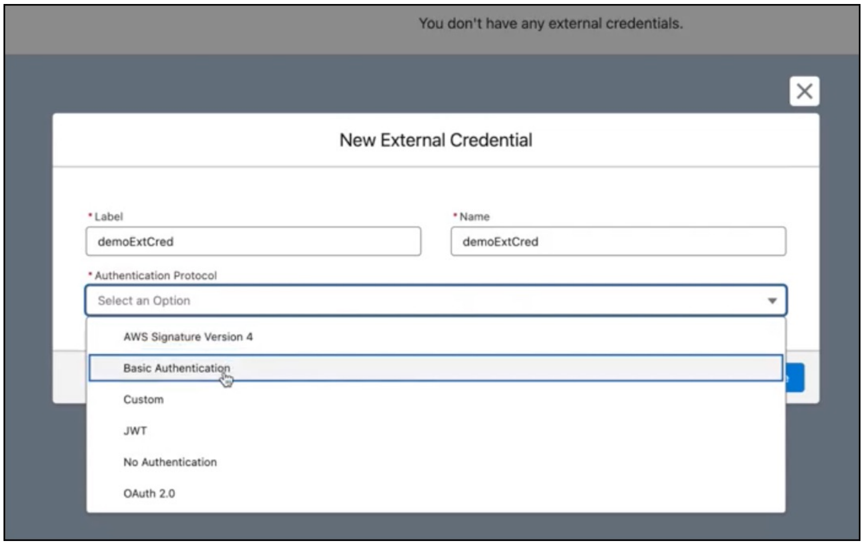
To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

1. Configure an External Credential

The external credential provides the external KMS the authentication to supply a key to your org.

1. In Setup, in the Quick Find box, enter `Named Credentials`, and then select **Named Credentials**.
2. Click **External Credentials**.
3. Enter a label and name for the external credential.
4. From the Authentication Protocol dropdown list, select a protocol type. See Authentication Protocols for Named Credentials.



You don't have any external credentials.

New External Credential

* Label: demoExtCred

* Name: demoExtCred

* Authentication Protocol: Select an Option

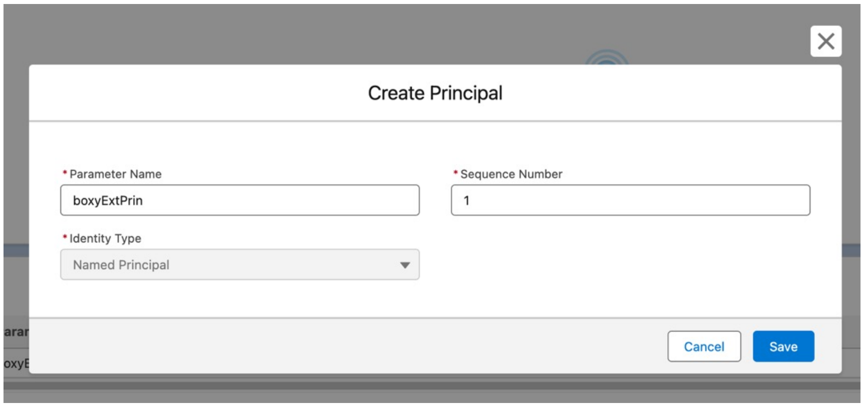
5. Save the new external named credential. Salesforce shows the properties page for your new named credential.

Leave the properties page open and then go on to configure an external named principal.

2. Configure an External Named Principal

The external named principal links an external credential to a permission set, so your org can make callouts by using the named credential.

1. If you aren't there already, open the properties page for the external credential for which you want to create a named principal.
2. In the Principals box, click **New**.
3. Enter a parameter name and leave the rest of the values as is.



Create Principal

* Parameter Name: boxyExtPrin

* Sequence Number: 1

* Identity Type: Named Principal

4. Save the new external named principal.

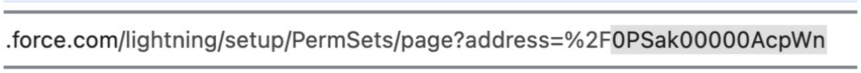
Next, create the linking permission set.

3. Create a Permission Set for the Named Principal

The members of the permission set can access the named principal.

Review Enable External Credential Principals for details on creating a permission set for a named principal.

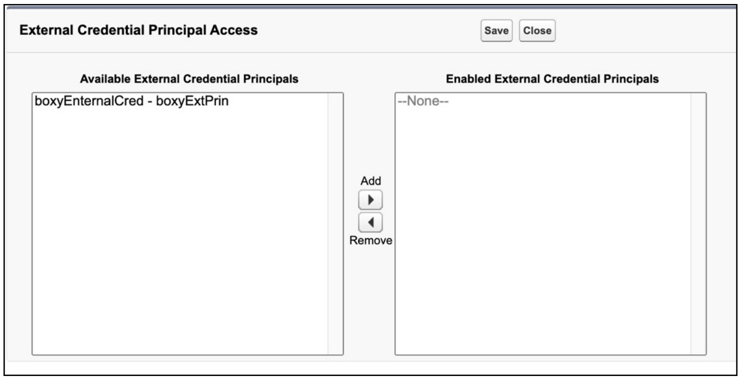
1. In Setup, in the Quick Find box, enter `Permission Sets`, and then select **Permission Sets**.
2. Select **New**.
3. Enter a label and an API name for the permission set.
4. Save the permission set.
   Salesforce shows the properties page for your new permission set.
5. While you're here, get the ID of the permission set from the browser address bar. You need the permission set ID later when you assign users.
   The permission set ID is everything to the right of `%2F` in the URL:



```
.force.com/lightning/setup/PermSets/page?address=%2F0PSak00000AcpWn
```

6. To show the principal access properties, select **External Credential Principal Access**.
7. In the External Credential Principal Access section, click **Edit**.

   Salesforce shows the external principal chooser.


External Credential Principal Access
Save
Close
Available External Credential Principals
boxyEnternalCred - boxyExtPrin
Add
Remove
Enabled External Credential Principals
--None--

8. Select the principal that you want to use, click **Add**, and then save your changes.

Next, assign the Automated Process user (`autoproc`) to the permission set.

4. Assign the autoproc User to the Permission Set

To assign the Automated Process user (`autoproc`) to the permission set, run a query on your org. You can use your preferred development environment. Always run a query to make this assignment, because you can't assign the `autoproc` user via the UI.

1. Open your preferred development environment that has access to your Salesforce org.
2. Prepare the query as shown in this example. In place of ***`permission_set_id`***, enter the permission set ID that you got when you created the permission set.

```
insert new PermissionSetAssignment(
  AssigneeId = [SELECT id FROM User where alias = 'autoproc'].Id,
  PermissionSetId = 'permission_set_id'
);
```

3. Execute the query.
   If your dev environment is set up properly, the result is `Success`.

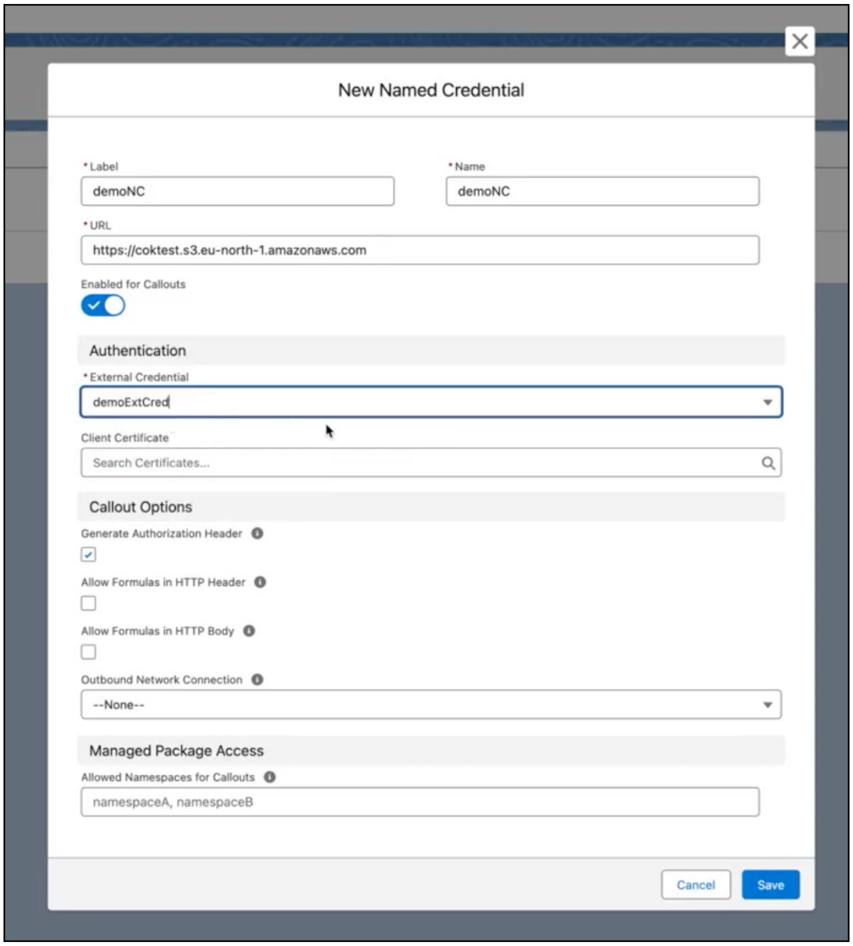
4. To verify the assignment, return to your permission set property page, and then click **Manage Assignments**.
   The Automated Process user is the only account assigned to the permission set.

Next, create the named credential.

5. Create a Named Credential for the Cache-Only Key

The named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition.

1. In Setup, in the Quick Find box, enter `Named Credentials` and then select **Named Credentials**.
2. Click **New**.
3. Enter values for the credential label and name.
4. In the URL field, enter the URL value that you saved earlier that locates the external key.
5. In the External Credentials field, enter the name of the external credential you created previously.



New Named Credential

* Label: demoNC
* Name: demoNC
* URL: https://coktest.s3.eu-north-1.amazonaws.com

Enabled for Callouts ✓

Authentication

* External Credential: demoExtCred

Client Certificate: Search Certificates...

Callout Options

Generate Authorization Header ☑

Allow Formulas in HTTP Header ☐

Allow Formulas in HTTP Body ☐

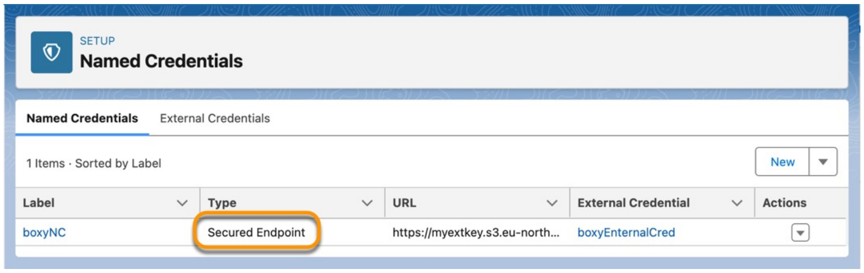
Outbound Network Connection: --None--

Managed Package Access

Allowed Namespaces for Callouts: namespaceA, namespaceB

   For guidance on the other New Named Credentials parameters, see Create or Edit an External Credential.

6. Save the new credential.
   In the Named Credentials list, your new credential has a type which isn't Legacy. (Named credentials with no named principal are Legacy named credentials.)



SETUP
Named Credentials

1 Items · Sorted by Label

Label	Type	URL	External Credential	Actions
boxyNC	Secured Endpoint	https://myextkey.s3.eu-north...	boxyEnternalCred	▾

Next, finish this process and create the cache-only key.

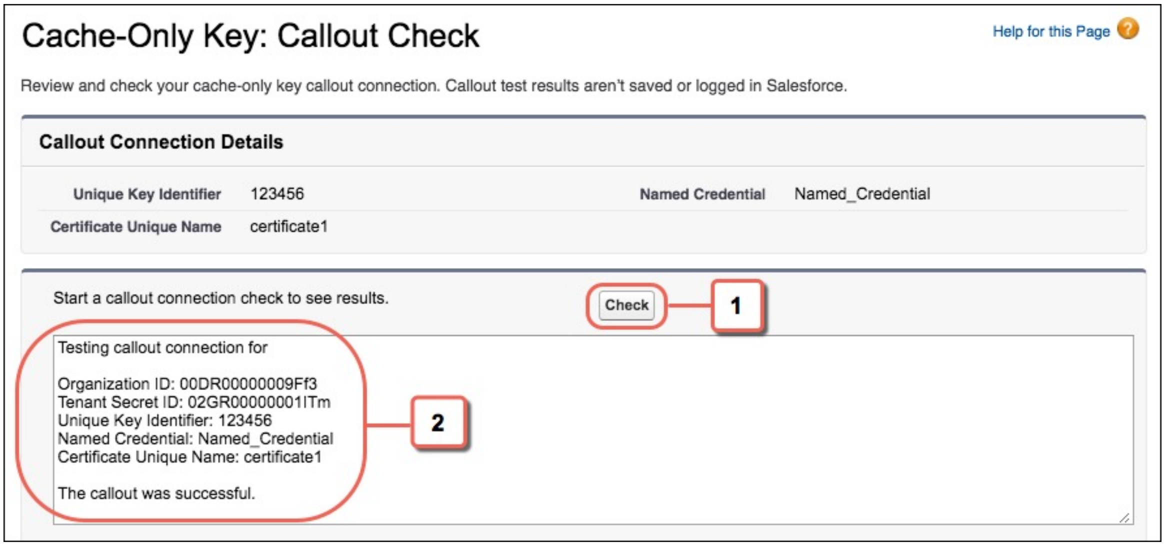
6. Use the Named Credential with a New Cache-Only Key

Define the cache-only key object that represents the external key.

1. In Setup, in the Quick Find box, enter `Key Management`, and then select **Key Management**.
2. Click **BYOK**.
   Salesforce shows the Bring Your Own Key page.

   Note: If you're asked for a certificate, create or select a self-signed or CA-signed certificate. See Generate a BYOK-Compatible Certificate.

3. From the Choose Certificate dropdown list, select a BYOK-compatible certificate.
4. Select **Use a Cache-Only Key**.
5. Enter the unique identifier for the external key as provided by the KMS that you created previously.
6. From the Named Credential dropdown list, select the named credential that you created earlier.



Cache-Only Key: Callout Check

Review and check your cache-only key callout connection. Callout test results aren't saved or logged in Salesforce.

Callout Connection Details

Unique Key Identifier	123456	Named Credential	Named_Credential
Certificate Unique Name	certificate1		

Start a callout connection check to see results.

```
Testing callout connection for

Organization ID: 00DR00000009Ff3
Tenant Secret ID: 02GR00000001ITm
Unique Key Identifier: 123456
Named Credential: Named_Credential
Certificate Unique Name: certificate1

The callout was successful.
```

Salesforce checks the connection to the endpoint specified by the named credential. If Salesforce can reach the endpoint, the key specified for the unique key identifier becomes the active key. All data marked for encryption by your encryption policy is encrypted with your cache-only key.

If Salesforce can't reach the specified endpoint, it displays an error to help you troubleshoot the connection.

7. When Salesforce can reach the endpoint, save your work.

Create and Assemble Your Key Material

The Shield Platform Encryption Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

Cache-only key material is wrapped in a JSON format. An example cache-only key is used throughout this article to illustrate how key material changes as you assemble it.

1. Generate a 256-bit AES data encryption key. You can use the cryptographically secure method of your choice.
2. Generate a 256-bit AES content encryption key by using a cryptographically secure method.
3. Generate and download your BYOK-compatible certificate.
4. Create the JWE protected header. The JWE protected header is a JSON object with three claims: the algorithm used to encrypt the content encryption key, the algorithm used to encrypt the data encryption key, and the unique ID of the cache-only key. Here's an example header to get us started.

   ```
   {"alg":"RSA-OAEP","enc":"A256GCM","kid":"982c375b-f46b-4423-8c2d-4d1a69152a0b"}
   ```

5. Encode the JWE protected header as BASE64URL(UTF8(JWE Protected Header)).

   ```
   eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00iLCJraWQiOiI5ODJjMzc1Yi1mNDZiLTQ0MjMtOGMy
   ZC00ZDFhNjkxNTJhMGIifQ
   ```

6. Encrypt the content encryption key with the public key from the BYOK certificate using the RSAES-OAEP algorithm. Then encode this encrypted content encryption key as BASE64URL(Encrypted CEK).

   ```
   l92QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvfT24oBCWkh6hy_dqAL7JlVO4
   49EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWWZ-sVK4pUw0B3lHwWBfpMsl4jf0exP5-5amiTZ5oP
   0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChH1wQjo12TQaWG_tiTwL1SgRd3YohuMVlmCdEmR2TfwTvryLPx4K
   bFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv46m8mUATZSD4hab8v3Mq4H3
   3CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFMlBC2Sd4yBKyjlDQKcSslCVav4buG8hkOJXY69iW_zhz
   tV3DoJJ90l-EvkMoHpw1llU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7IvzeneL5I81QjQlDJmZhbLLor
   FHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LOeOMkCFl-9PwrULWyRTLMI7CdZIm7jb8v9AL
   xCmDgqUi1yvEeBJhgMLezAWtxvGGkejc0BdsbWaPFXlI3Uj7C-Mw8LcmpSLKZyEnhj2x-3Vfv5hIVauC6ja1
   B6Z_UcqXKOc
   ```

7. Generate an initialization vector for use as input to the data encryption key's AES wrapping. Then encode it in base64url.

   ```
   N2WVMbpAxipAtG9O
   ```

8. Wrap your data encryption key with your content encryption key.
   1. **a.** Encode the JWE header as ASCII(BASE64URL(UTF8(JWE Protected Header))).
   2. **b.** Reform authenticated encryption on the data encryption key with the AES GCM algorithm. Use the content encryption key as the encryption key, the initialization vector (the bytes, not the base64 URL encoded version), and the Additional Authenticated Data value, requesting a 128-bit Authentication Tag output.
   3. **c.** Encode the resulting ciphertext as BASE64URL(Ciphertext).
   4. **d.** Encode the Authentication Tag as BASE64URL(Authentication Tag).

   ```
   63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk32DinS_zFo4
   ```

   and

   ```
   HC7Ev5lmsbTgwyGpeGH5Rw
   ```

9. Assemble your JWE as a compact serialization of all the preceding values. Concatenate values separated by a period.

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00iLCJraWQiOiI5ODJjMzc1Yi1mNDZiLTQ0MjMtOGMy
ZC00ZDFhNjkxNTJhMGIifQ.l92QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvf
T24oBCWkh6hy_dqAL7JlVO449EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWWZ-sVK4pUw0B3lHwWB
fpMsl4jf0exP5-5amiTZ5oP0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChH1wQjo12TQaWG_tiTwL1SgRd3Yoh
uMVlmCdEmR2TfwTvryLPx4KbFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv
46m8mUATZSD4hab8v3Mq4H33CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFMlBC2Sd4yBKyjlDQKcSsl
CVav4buG8hkOJXY69iW_zhztV3DoJJ90l-EvkMoHpw1llU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7Iv
zeneL5I81QjQlDJmZhbLLorFHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LOeOMkCFl-9Pwr
ULWyRTLMI7CdZIm7jb8v9ALxCmDgqUi1yvEeBJhgMLezAWtxvGGkejc0BdsbWaPFXlI3Uj7C-Mw8LcmpSLKZ
yEnhj2x-3Vfv5hIVauC6ja1B6Z_UcqXKOc.N2WVMbpAxipAtG9O.63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk
32DinS_zFo4.HC7Ev5lmsbTgwyGpeGH5Rw
```

For more detailed examples of this process, check out the sample Cache-Only Key Wrapper in Github. You can use either the utility in this repository or another service of your choosing.

Add Replay Detection for Cache-Only Keys

Replay detection protects your cache-only keys if a callout is fraudulently intercepted. When enabled, replay detection inserts an autogenerated, unique marker called a RequestIdentifier into every callout. The RequestIdentifier includes the key identifier, a nonce generated for that callout instance, and the nonce required from the endpoint. The RequestIdentifier serves as a random, one-time identifier for each valid callout request. After you set up your key service to accept and return the RequestIdentifier, any callout with missing or mismatched RequestIdentifiers is aborted.

1. Update your key service to extract the nonce generated for the callout instance from the RequestIdentifier. Here's what the nonce looks like.

   ```
   e5ab58fd2ced013f2a46d5c8144dd439
   ```

2. Echo this nonce in the JWE protected header, along with the algorithm used to encrypt the content encryption key, the algorithm used to encrypt the data encryption key, and the unique ID of the cache-only key. Here's an example.

   ```
   {"alg":"RSA-OAEP","enc":"A256GCM","kid":"982c375b-f46b-4423-8c2d-4d1a69152a0b","jti":"e5ab58fd2ced013f2a46d5c8144dd439"}
   ```

3. From Setup, in the Quick Find box, enter *`Encryption Settings`*, and then click **Encryption Settings**.

4. In the Advanced Encryption Settings section, turn on **Enable Replay Detection for Cache-Only Keys**.

   You can also enable replay detection programmatically. For more information, see EncryptionKeySettings in the *Metadata API Developer Guide*.

   From now on, every callout to an external key service includes a unique RequestIdentifier.

Warning: If you enable replay detection but don't return the nonce with your cache-only key material, Salesforce aborts the callout connection and displays a POTENTIAL_REPLAY_ATTACK_DETECTED error.

USER PERMISSIONS

To create, edit, and delete named credentials:

- Customize Application

To allow cache-only keys with BYOK:

- Customize Application

  AND

  Manage Encryption Keys

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Check Your Cache-Only Key Connection

Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

The Cache-Only Key: Callout Check page is accessible after you enable the Cache-Only Key Service in your org and make your first callout. Data presented as part of a callout check are never stored in the system of record.

1. From Setup, enter `Platform Encryption` in the Quick Find box, then select **Key Management**.
2. Choose the Certificate Unique Name and Named Credential associated with your Unique Key Identifier.
3. In the Actions column, next to the key material you want to check, click **Details**.
4. On the Cache-Only Key: Callout Check page, click **Check**.
   Details about your callout connection display on the page. It can take a few moments for the callout check to complete and display the results.

EDITIONS

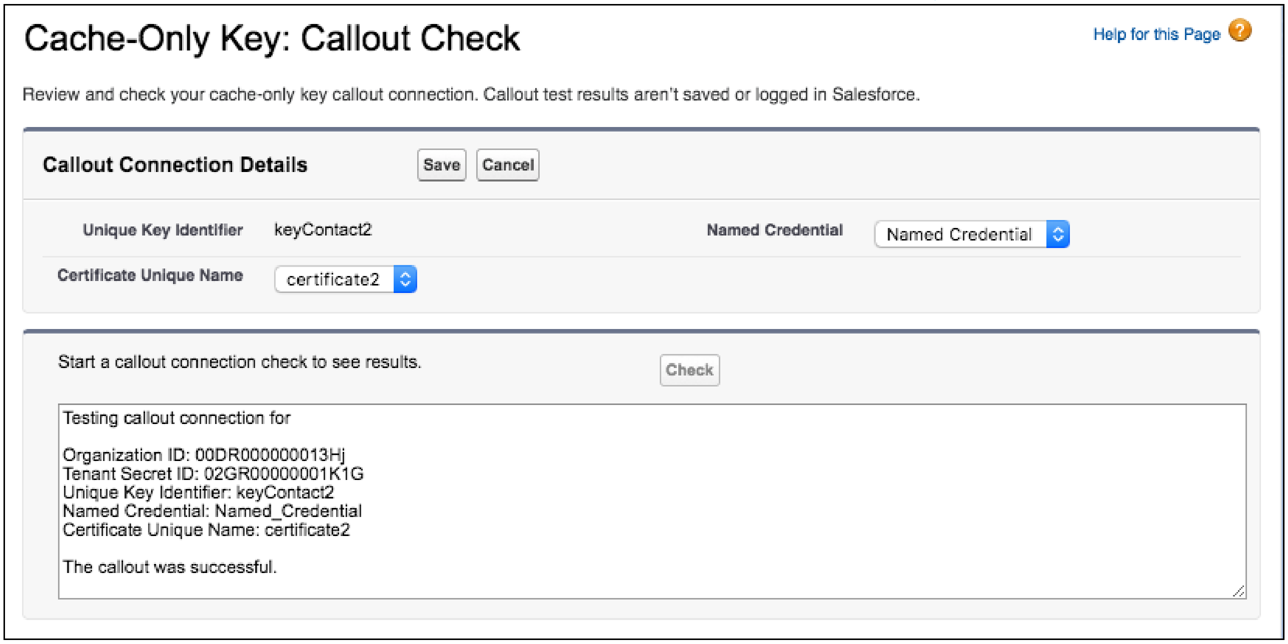
Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys



Cache-Only Key: Callout Check

Review and check your cache-only key callout connection. Callout test results aren't saved or logged in Salesforce.

Callout Connection Details

Unique Key Identifier: keyContact2

Named Credential: Named Credential

Certificate Unique Name: certificate2

Start a callout connection check to see results.

```
Testing callout connection for

Organization ID: 00DR000000013Hj
Tenant Secret ID: 02GR00000001K1G
Unique Key Identifier: keyContact2
Named Credential: Named_Credential
Certificate Unique Name: certificate2

The callout was successful.
```

5. Review the details about your callout connection. If your callout connection was unsuccessful, you see a descriptive error message at the bottom of the results pane. Use this message to make the appropriate adjustments to your key service.

Destroy a Cache-Only Key

When you destroy a cache-only key, you're destroying two things: the key in the cache and the callout connection to the key service.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. In the Key Management Table, select a key type.
3. Find your key in the table and click **Destroy**.
   Your key material's status is changed to Destroyed, and callouts to this key stop. Data encrypted with this key material is masked with "?????" in the app.

Note: Your cache-only key is unique to your org and to the specific data to which it applies. When you destroy a cache-only key, related data isn't accessible unless you reactivate it and make sure that Salesforce can fetch it.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Reactivate a Cache-Only Key

If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup or programmatically through the API. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

1. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
2. Find your key in the table and click **Activate**.
   The Shield Key Management Service fetches the reactivated cache-only key from your key service and uses it to access data that was previously encrypted with it.

   Note: You can sync your data to your active cache-only key just like you can with any other key material.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and the Cache-Only Key Service.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Considerations for Cache-Only Keys

These considerations apply to all data that you encrypt using the Shield Platform Encryption Cache-Only Key Service.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Named Credentials

To use named principals with the Shield Platform Encryption Cache-Only Keys, create a permission set for external credential principal access, and assign that permission set to the `autoproc` user. See Use a Named Principal-Based Credential for a Cache-Only Key.

Retry Policy

If Salesforce can't reach your external key service, the callout fails and your active cache-only key's status is set to Destroyed. This policy prevents excessive loads on both services. The Cache-Only Key Service then periodically retries the callout to help you minimize down time. Retries occur one time per minute for five minutes, then one time every five minutes for 24 hours. If the Cache-Only Key Service can successfully complete a callout during this retry period, your cache-only key's status is reset to Active.

At any point during a retry period, you can activate your key material through Setup or the API pending remote key service availability. If you reactivate your key material during the retry period, all retry attempts stop.

The RemoteKeyCalloutEvent object captures every callout to your key service. You can subscribe to this event with after insert Apex triggers, and set up real-time alerts that notify you when a callout fails.

401 HTTP Responses

If there's a 401 HTTP response, Salesforce automatically refreshes any OAuth token associated with your named credential, and retries the request.

CRM Analytics

Backups of CRM Analytics data are encrypted with your Shield Platform Encryption keys. If you encrypt data in CRM Analytics datasets with a cache-only key, make sure that the Analytics cache-only key is in the same state as your Fields and Files (Probabilistic) cache-only key.

Setup Audit Trail

Setup Audit Trail records activated cache-only key versions differently depending on whether a cache-only key with the Active status exists when you reactivate the key.

However, if you reactivate a destroyed key and there's already another key with the Active status, the Setup Audit Trail shows the reactivated key with an updated version number.

Cache-Only Keys and Key Types

Use a separate cache-only key for each type of data you want to encrypt. You can't use a cache-only key with multiple key types. For example, you can't use a cache-only key to encrypt both search indexes and CRM Analytics data.

Service Protections

To protect against Shield KMS interruptions and ensure smooth encryption and decryption processes, you can have up to 10 active and archived cache-only keys of each type.

If you reach your key limit, destroy an existing key so that you can create, upload, reactivate, rearchive, or create a callout to another one. Remember to synchronize your data with an active key before destroying key material.

Hyperforce Migration

When your org moves from a non-Hyperforce platform to Hyperforce, you may need to revisit your AWS KMS IP connection settings. We recommend that Hyperforce customers adopt the best practices listed in the topic Preferred Alternatives to IP Allowlisting on Hyperforce as soon as possible.

Troubleshoot Cache-Only Keys

One or more of these frequently asked questions can help you troubleshoot any problems that arise with Shield Platform Encryption's Cache-Only Key Service.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

The callout to my key service isn't going through. What can I do?

Callouts can fail for various reasons. Review the error message that displays and follow these tips for resolving the problem. All callouts are recorded in the RemoteKeyCalloutEvent object.

Table 4: Cache-Only Key Service Errors and Status Codes

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
AUTHENTICATION_FAILURE_RESPONSE	Authentication with the remote key service failed with the following error: {error}.	Check the authentication settings for your chosen named credential.
DESTROY_HTTP_CODE	The remote key service returned an HTTP error: {000}. A successful HTTP response returns a 200 code.	To find out what went wrong, review the HTTP response code.
EMPTY_RESPONSE	The remote key service callout returned an empty response. Contact your remote key service for help.	Contact your remote key service.
ERROR_HTTP_CODE	The remote key service returned an unsupported HTTP response code: {000}. A successful HTTP response returns a 200 code.	To find out what went wrong, review the HTTP response code.
ILLEGAL_PARAMETERS_IN_JWE_HEADER	Your JWE header must use {0}, but no others. Found: {1}.	Remove the unsupported parameters from your JWE header.
INCORRECT_ALGORITHM_IN_JWE_HEADER	The remote key service returned a JWE header that	The algorithm for encrypting the content encryption key in

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
	specified an unsupported algorithm (alg): {algorithm}.	your JWE header must be in RSA-OAEP format.
INCORRECT_DATA_ENCRYPTION_KEY_SIZE	Data encryption keys encoded in a JWE must be 32 bytes. Yours is {value} bytes.	Make sure that your data encryption key is 32 bytes.
INCORRECT_ENCRYPTION_ALGORITHM_IN_JWE_HEADER	The remote key service returned a JWE header that specified an unsupported encryption algorithm (enc): {your enc}.	The algorithm for encrypting the data encryption key in your JWE header must be in A256GCM format.
INCORRECT_KEYID_IN_JSON	The remote key service returned JSON with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
INCORRECT_KEYID_IN_JWE_HEADER	The remote key service returned a JWE header with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
MALFORMED_CONTENT_ENCRYPTION_KEY	The remote key service returned a content encryption key in the JWE that couldn't be decrypted with the certificate's private key. Either the JWE is corrupted, or the content encryption key is encrypted with a different key.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
MALFORMED_DATA_ENCRYPTION_KEY	The content encryption key couldn't decrypt the data encryption key that was returned in the remote key service's JWE. The data encryption key is either malformed, or encrypted with a different content encryption key.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate. Named credentials must call out to an HTTPS endpoint.
MALFORMED_JSON_RESPONSE	We can't parse the JSON returned by your remote key service. Contact your remote key service for help.	Contact your remote key service.
MALFORMED_JWE_RESPONSE	The remote key service returned a malformed JWE token that can't be decoded. Contact your remote key service for help.	Contact your remote key service.
MISSING_PARAMETERS_IN_JWE_HEADER	Your JWE header is missing one or more parameters. Required: {0}. Found:{1}.	Make sure that your JWE header includes all required values. For example, if Replay Detection is enabled, the JWE header must include the nonce value extracted from the cache-only key callout.
POTENTIAL_REPLAY_ATTACK_DETECTED	The remote key service returned a JWE header with an incorrect nonce value. Expected: {0}. Actual: {1}	Make sure that your JWE header includes the RequestID included in the callout.

RemoteKeyCalloutEvent Status Code	Error	Tips for Fixing the Problem
ACCESS TO NC DENIED	We couldn't access the credential. You don't havethe required permissions, or the external credential you specified doesn't exist.	Make sure that you specified the correct named credential. Also, this error occurs if you haven't added the autoproc user to the external credential principal permission set. See Use a Named Principal-Based Credential for a Cache-Only Key.
RESPONSE_TIMEOUT	The remote key service callout took too long and timed out. Try again.	If your key service is unavailable after multiple callout attempts, contact your remote key service.
UNKNOWN_ERROR	The remote key service callout failed and returned an error: {000}.	Contact your remote key service.
UNKNOWN_ERROR	The remote key service callout failed and returned an error: java.security.cert.CertificateExpiredException: NotAfter: {date and time of expiration}	The certificate for your cache-only key has expired. Update your cache-only key material to use an active BYOK-compatible certificate.
UNKNOWN_EXCEPTION: Urgent	Your Cache-Only key is unavailable.	Refer to the "UNKNOWN_EXCEPTION: Urgent" information later on this page.

The following key service errors can prevent the callout from completing. If you see errors related to these problems, contact your key service administrator for help.

- The JWE is corrupt or malformed.
- The data encryption key is malformed.
- The key service returned a malformed JWE token.
- The key service returned an empty response.

For uniform resource use, Salesforce limits the amount of time for each key service callout to 3 seconds. If the callout takes more than the allotted time, Salesforce fails the callout with a timeout error. Check that your key service is available. Make sure that your named credential references the correct endpoint—check the URL, including the IP address.

Can I execute a remote callout in Apex?

Yes. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to. To reference a named credential from a callout definition, use the named credential URL. A named credential URL contains the scheme callout, the name of the named credential, and an optional path. For example: callout:My_Named_Credential/some_path.

See Named Credentials as Callout Endpoints in the Apex Developer Guide.

Can I monitor my callout history?

If you want to review or track cache-only key events, use the RemoteKeyCalloutEvent standard object. Either use the `describeSObjects()` call to view event information, or an after insert Apex trigger to perform custom actions after each callout. For example, you can write a trigger that stores `RemoteKeyCallout` events in a custom object. When you store `RemoteKeyCallout` events in a custom object, you can monitor your callout history. See the RemoteKeyCalloutEvent entry in the *Salesforce Object Reference* for more information.

The Setup Audit Trail tracks changes in key material state and named credential settings. Callout history isn't recorded in log files.

I see "?????", !!!!!, 08/08/1888, or 01/01/1777 instead of my data when I try to access data encrypted with a cache-only key, Why?

The value that you see is a string reserved for masking notifications. The presence of a reserved masked value means one of two things. Either the connection to your key service is broken and we can't fetch your key, or the data is encrypted with a destroyed key. Check that your key service is available and that your named credential references the correct endpoint. If any key versions are marked as Destroyed as a result of a key service failure, recover the connection and activate the key version by hand. The topic Why Isn't My Encrypted Data Masked? on page 32 lists all the reserved masking notification strings.

I see either "????? ?????" or the error "UNKNOWN_EXCEPTION, Urgent: your key service unavailable. You can't edit, view, or create encrypted records without the encryption key provided by this service. Contact your Salesforce security admin." whenever I open records that contain previously encrypted data, Why?

This error can result if your Cache-Only key Key Management Server is unavailable. If you're confident that your cache-only key exists, check that the connections from AWS to Hyperforce are allowed. Your AWS KMS must permit access to the required the Salesforce Hyperforce IP addresses.

We recommend that Hyperforce customers adopt best practices as documented in the topic Preferred Alternatives to IP Allowlisting on Hyperforce.

My certificate is about to expire. What do I do?

An expired certificate doesn't affect the active state of the secret that it wraps. Your certificate gives assurance to the recipient that the received secret was sent and wrapped by you. If you use an expired certificate, your secret is still protected, but the receiving party is notified that the certificate is expired. Salesforce does not block your secret if it's wrapped with an expired certificate.

Do I have to make a new named credential every time I rotate a key?

Nope. You can use a named credential with multiple keys. As long as you host your key material at the endpoint specified in an existing named credential, you're all set. When you rotate your key material, change the key ID in the Unique Key Identifier field. Double-check that your new key is stored at the specified endpoint URL in your named credential.

Can I use legacy named credentials with cache-only keys?

Yes. You can use whichever type is supported by your external key service.

I'm still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive or Salesforce Customer Support. They'll put you in touch with a support team specific to this feature.

Configure Your Cache-Only Key Callout Connection

Use a named credential to specify the endpoint for your callout, and identify the key that you want to fetch from your endpoint.

Note: Some endpoints support legacy named credentials, and others require named principal-based named credentials. This topic doesn't show you how to configure a named principal-based credential. See Use a Named Principal-Based Credential for a Cache-Only Key.

1. Make sure that your org has an active Fields and Files (Probabilistic) key, either Salesforce-generated or customer-supplied.
   - From Setup, in the Quick Find box, enter **Encryption Settings**, and then select **Encryption Settings**. Turn on **Generate Initial Probabilistic Tenant Secret**.
   - From Setup, in the Quick Find box, enter `Key Management`, and then select **Key Management**. Select the **Fields and Files (Probabilistic)**tab, and then click **Generate Tenant Secret**.
2. From Setup, in the Quick Find box, enter `Named Credential`, and then select **Named Credential**.

   Tip: A named credential provides an authenticated callout mechanism through which Salesforce can fetch your key material. Because named credentials are allowlisted, they're a secure and convenient channel for key material stored outside of Salesforce.

   Learn more about named credentials, how to define a named credential, and how to grant access to authentication settings for named credentials in Salesforce Help.
3. Create a named credential. Specify an HTTPS endpoint from which Salesforce can fetch your key material.
4. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
5. In the Advanced Encryption Settings section, turn on **Allow Cache-Only Keys**.

   You can also enable the Cache-Only Key Service programmatically. For more information, see EncryptionKeySettings in the *Metadata API Developer Guide*.

   Note: If you turn off **Allow Cache-Only Keys**, data that's encrypted with cache-only key material remains encrypted and Salesforce continues to invoke secured callouts. However, you can't modify your cache-only key configuration or add new ones. If you don't want to use cache-only keys, rotate your key material to use customer-supplied (BYOK) key material. Then synchronize all your data, and turn off **Allow Cache-Only Keys**.
6. From Setup, in the Quick Find box, enter `Platform Encryption`, and then select **Key Management**.
7. In the Key Management Table, select a key type.
8. Click **Bring Your Own Key**.
9. Select a BYOK-compatible certificate from the Choose Certificate dropdown.
10. Select **Use a Cache-Only Key**.
11. For Unique Key Identifier, enter your KID—the unique key identifier for your data encryption key. Your identifier can be a number, a string (2018_data_key), or a UUID (982c375b-f46b-4423-8c2d-4d1a69152a0b).

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption, and Cache-Only Keys.

USER PERMISSIONS

To create, edit, and delete named credentials:
- Customize Application

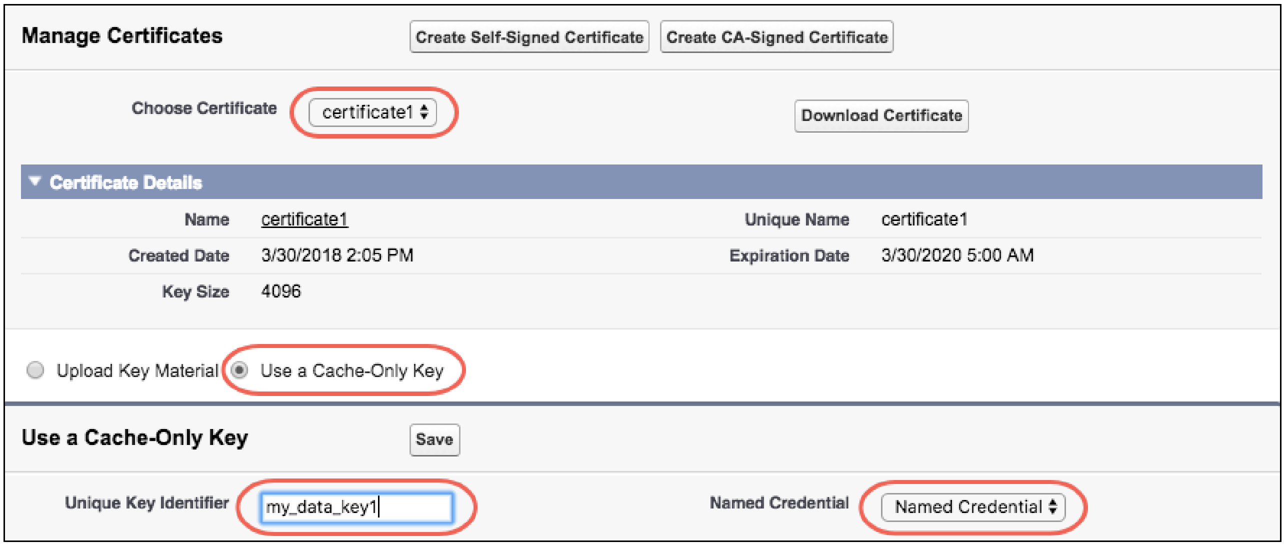
To allow cache-only keys with BYOK:
- Customize Application

  AND

  Manage Encryption Keys

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:
- Manage Encryption Keys

12. In the Named Credential dropdown, select the named credential associated with your key. You can have multiple keys associated with each named credential.



Choose Certificate: certificate1

Name	certificate1	Unique Name	certificate1
Created Date	3/30/2018 2:05 PM	Expiration Date	3/30/2020 5:00 AM
Key Size	4096		

○ Upload Key Material ● Use a Cache-Only Key

Unique Key Identifier: my_data_key1

Named Credential: Named Credential

Salesforce checks the connection to the endpoint specified by the named credential. If Salesforce can reach the endpoint, the key specified for the Unique Key Identifier becomes the active key. All data marked for encryption by your encryption policy is encrypted with your cache-only key.

If Salesforce can't reach the specified endpoint, an error displays to help you troubleshoot the connection.

Cache-only key status is recorded as Fetched on the Key Management page. In Enterprise API, the TenantSecret `Source` value is listed as Remote.

Tip: You can monitor key configuration callouts in the Setup Audit Trail. When a callout to an active or archived cache-only key is successful, the Setup Audit Trail logs an Activated status. Individual callouts aren't monitored in Setup Audit Trail.

Shield Platform Encryption Customizations

Some features and settings require adjustment before they work with encrypted data.

Apply Encryption to Fields Used in Matching Rules

Matching rules used in duplicate management help you maintain clean and accurate data. To make fields encrypted with Shield Platform Encryption compatible with standard and custom matching rules, use the deterministic encryption scheme.

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. Shield Platform Encryption is compatible with several operators and functions, and can render encrypted data in text, date, and date/time formats, and reference quick actions.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Apply Encryption to Fields Used in Matching Rules

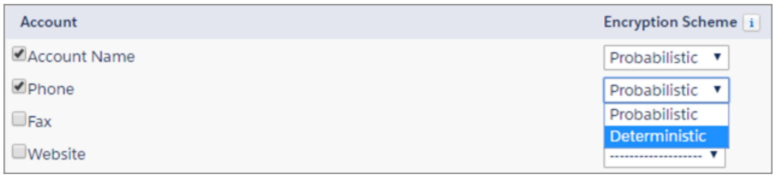
Matching rules used in duplicate management help you maintain clean and accurate data. To make fields encrypted with Shield Platform Encryption compatible with standard and custom matching rules, use the deterministic encryption scheme.

Before you start, turn on **Deterministic Encryption** from the Encryption Settings page. If you don't have a Fields (Deterministic) type tenant secret, create one from the Key Management page.

Important: Matching rules used in duplicate management don't support probabilistically encrypted data.

Follow these steps to add encrypted fields to existing custom matching rules.

1. From Setup, in the Quick Find box, enter `Matching Rules`, and then select **Matching Rules**.
2. Deactivate the matching rule that reference fields that you want to encrypt. If your matching rule is associated with an active duplicate rule, first deactivate the duplicate rule from the Duplicate Rules page. Then return to the Matching Rules page and deactivate the matching rule.
3. From Setup, in the Quick Find box, enter `Encryption Settings`, and then select **Encryption Settings**.
4. In the Advanced Encryption Settings section, click **Select Fields**.
5. Click **Edit**.
6. Select the fields that you want to encrypt, and select **Deterministic** from the Encryption Scheme list.



Account	Encryption Scheme
☑ Account Name	Probabilistic ▼
☑ Phone	Probabilistic ▼ Probabilistic Deterministic
□ Fax	
□ Website	------------------ ▼

7. Save your work.

   Tip: Standard matching rules are automatically deactivated when encryption is added to a field referenced by that rule. To encrypt fields referenced in standard matching rules, follow steps 3–8.

8. After you get the email verifying encryption's been enabled on your fields, reactivate your matching rule and associated duplicate management rule.
   Matching rules used in duplicate management now return exact and fuzzy matches on encrypted data.

Example: Let's say that you encrypted the Billing Address on your Contacts, and you want to add this field to a custom matching rule. First, deactivate the rule or rules that you want to add this field to. Make sure that the Billing Address field is encrypted with the deterministic encryption scheme. Then add Billing Address to your custom matching rule, just like how you add any other field. Finally, reactivate your rule.

When you rotate your key material, you must update custom matching rules that reference encrypted fields. After you rotate your key material, deactivate and then reactivate the affected matching rules. Then contact Salesforce to request the background encryption process. When the background encryption process finishes, your matching rules can access all data encrypted with your active key material.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

USER PERMISSIONS

To view setup:
- View Setup and Configuration

To enable encryption key (tenant secret) management:
- Manage Profiles and Permission Sets

Important: To ensure accurate matching results, customers who used the beta version of this feature must deactivate any matching rules that reference encrypted fields and then reactivate them. If your custom matching rule fails on reactivation, contact Salesforce for help with reactivating your match index.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. Shield Platform Encryption is compatible with several operators and functions, and can render encrypted data in text, date, and date/time formats, and reference quick actions.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Available in Salesforce Classic.

Note: Formula fields that reference encrypted data are supported only in Salesforce Classic. They aren't supported in Lightning Experience or via SOQL. If you work exclusively in Lightning Experience or have dependencies on formula fields that require Lightning Experience, we recommend that you don't reference encrypted fields in formulas. The following examples apply to Salesforce Classic.

Supported Operators, Functions, and Actions

Supported operators and functions:

- `&` and `+` (concatenate)
- `BLANKVALUE`
- `CASE`
- `HYPERLINK`
- `IF`
- `IMAGE`
- `ISBLANK`
- `ISNULL`
- `NULLVALUE`

Also supported:

- Spanning
- Quick actions

Formulas can return data only in `text`, `date`, or `date/time` formats.

`&` and `+` (Concatenate)

This works:

```
(encryptedField__c & encryptedField__c)
```

Why it works:

This formula works because `&` is supported.

This doesn't work:

```
LOWER(encryptedField__c & encryptedField__c)
```

Why it doesn't work:

`LOWER` isn't a supported function, and the input is an encrypted value.

Case

`CASE` returns encrypted field values, but doesn't compare them.

This works:

```
CASE(custom_field__c, "1", cf2__c, cf3__c))
```

where either or both `cf2__c` and `cf3__c` are encrypted

Why it works:

`custom_field__c` is compared to "1". If it's true, the formula returns `cf2__c` because it's not comparing two encrypted values.

This doesn't work:

```
CASE("1", cf1__c, cf2__c, cf3__c)
```

where `cf1__c` is encrypted

Why it doesn't work:

You can't compare encrypted values.

`ISBLANK` and `ISNULL`

This works:

```
OR(ISBLANK(encryptedField__c), ISNULL(encryptedField__c))
```

Why it works:

Both `ISBLANK` and `ISNULL` are supported. `OR` works in this example because `ISBLANK` and `ISNULL` return a Boolean value, not an encrypted value.

Spanning

This works:

```
(LookupObject1__r.City & LookupObject1__r.Street) &
 (LookupObject2__r.City & LookupObject2__r.Street) &
  (LookupObject3__r.City & LookupObject3__r.Street) &
   (LookupObject4__r.City & LookupObject4__r.Street)
```

How and why you use it:

Spanning retrieves encrypted data from multiple entities. For example, let's say you work in the customer service department for Universal Containers. A customer has filed a case about a distribution problem, and you want to see the scope of the issue. You want all the shipping addresses related to this particular case. This example returns all the customers' shipping addresses as a single string in your case layout.

Validation

The encryption validation service checks your org to make sure that it's compatible with encrypted formula field types.

When you encrypt a given field, the validation service:

- Retrieves all formula fields that reference the field
- Verifies that the formula fields are compatible with encryption
- Verifies that the formula fields aren't used elsewhere for filtering or sorting

Limits

Up to 200 formula fields can reference a given encrypted custom field. A field that is referenced by more than 200 formula fields can't be encrypted. If you must reference an encrypted custom field from more than 200 formula fields, contact Salesforce.

When you specify multiple fields to encrypt at one time, the 200-field limit is applied to the whole batch. If you know that you're encrypting fields that have multiple formula fields pointing to them, encrypt those fields one at a time.

Tradeoffs and Limitations of Shield Platform Encryption

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Shield Platform Encryption Best Practices

Take the time to identify the most likely threats to your org. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

Considerations for Using Deterministic Encryption

These considerations apply to data encrypted with Shield Platform Encryption's deterministic encryption scheme. Some considerations manifest differently depending on whether data is encrypted with the case-sensitive or case-insensitive deterministic encryption scheme.

Shield Platform Encryption and the Lightning Experience

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

Field Limits with Shield Platform Encryption

It's good practice to use validation rules to enforce these field limits. In addition, because encrypted content is often longer than its ciphertext, encrypting a field can impose further limits on the values that you store in that field. Therefore, test your field limits in longer fields, such as Address and Subject, and on any encrypted field that contains non-ASCII values such as Chinese, Japanese, or Korean-encoded data.

Which Salesforce Apps Don't Support Shield Platform Encryption?

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

Shield Platform Encryption Best Practices

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Take the time to identify the most likely threats to your org. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

1. Define a threat model for your organization.

   To identify the threats that are most likely to affect your organization, walk through a formal threat modeling exercise. Use your findings to create a data classification scheme, which can help you decide what data to encrypt.

2. Encrypt only where necessary.

   - Not all data is sensitive. Focus on information that requires encryption to meet your regulatory, security, compliance, and privacy requirements. Unnecessarily encrypting data impacts functionality and performance.
   - Evaluate your data classification scheme early and work with stakeholders in security, compliance, and business IT departments to define requirements. Balance business-critical functionality against security and risk measures and challenge your assumptions periodically.

3. Create a strategy early for backing up and archiving keys and data.

   If your tenant secrets are destroyed, reimport them to access your data. You are solely responsible for making sure that your data and tenant secrets are backed up and stored in a safe place. Salesforce cannot help you with deleted, destroyed, or misplaced tenant secrets.

4. Read the Shield Platform Encryption considerations and understand their implications on your organization.

   - Evaluate the impact of the considerations on your business solution and implementation.
   - Test Shield Platform Encryption in a sandbox environment before deploying to a production environment. Encryption policy settings can be deployed using change sets.
   - Before enabling encryption, fix any violations that you uncover. For example, if you reference encrypted fields in a SOQL ORDER BY clause, a violation occurs. Fix the violation by removing references to the encrypted fields.
   - When requesting feature enablement, such as pilot features, give Salesforce Customer Support several days lead time. The time to complete the process varies based on the feature and how your org is configured.

5. Analyze and test AppExchange apps before deploying them.

   - If you use an app from the AppExchange, test how it interacts with encrypted data in your organization and evaluate whether its functionality is affected.
   - If an app interacts with encrypted data that's stored outside of Salesforce, investigate how and where data processing occurs and how information is protected.
   - If you suspect Shield Platform Encryption could affect the functionality of an app, ask the provider for help with evaluation. Also discuss any custom solutions that must be compatible with Shield Platform Encryption.
   - Apps on the AppExchange that are built exclusively using Lightning Platform inherit Shield Platform Encryption capabilities and limitations.

6. Use out-of-the-box security tools.

   Shield Platform Encryption is not a user authentication or authorization tool. To control which users can see which data, use out-of-the-box tools such as field-level security settings, page layout settings, and sharing rules, rather than Shield Platform Encryption.

7. Grant the Manage Encryption Keys user permission to authorized users only.

   Users with the Manage Encryption Keys permission can generate, export, import, and destroy organization-specific keys. Monitor the key management activities of these users regularly with the setup audit trail.

8. Synchronize your existing data with your active key material.

   Existing field and file data is not automatically encrypted when you turn on Shield Platform Encryption. To encrypt existing field data, update the records associated with the field data. This action triggers encryption for these records so that your existing data is encrypted at rest. To encrypt existing files or get help updating other encrypted data, contact Salesforce. We can encrypt existing file data in the background to ensure data alignment with the latest encryption policy and key material.

   When you contact Salesforce support to request the background encryption service, allow at least a week before you need the background encryption completed. The time to complete the process varies based on the volume of data involved. It could take several days.

9. Handle currency and number data with care.

   Currency and Number fields can't be encrypted because they could have broad functional consequences across the platform, such as disruptions to roll-up summary reports, report timeframes, and calculations. You can often keep private, sensitive, or regulated data of this variety safe in other encryption-supported field types.

10. Communicate to your users about the impact of encryption.

    Before you enable Shield Platform Encryption in a production environment, inform users about how it affects your business solution. For example, share the information described in Shield Platform Encryption considerations, where it's relevant to your business processes.

11. Encrypt your data using the most current key.

    When you generate a new tenant secret, any new data is encrypted using this key. However, existing sensitive data remains encrypted using previous keys. In this situation, Salesforce strongly recommends re-encrypting these fields using the latest key. Contact Salesforce for help with re-encrypting your data.

12. Use discretion when granting login as access to users or Salesforce Customer Support.

    If you grant login access to a user, and they have field level security access to an encrypted field, that user is able to view encrypted data in that field in plaintext.

    If you want Salesforce Customer Support to follow specific processes around asking for or using login as access, you can create special handling instructions. Salesforce Customer Support follows these instructions in situations where login as access may help them resolve your case. To set up these special handling instructions, contact your account executive.

General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Leads

Lead and Case assignment rules, workflow rules, and validation rules work normally when Lead fields are encrypted. Matching and de-duplication of records during lead import works with deterministic encryption but not probabilistic encryption. Einstein Lead Scoring isn't available.

Apex Lead Conversion works normally, but PL-SQL-based lead conversion isn't supported.

User Email

Many Salesforce features rely on the User Email field. The following products and features behave differently when User Email is encrypted.

- If the Email field on the User object is encrypted with field-level encryption, you don't receive critical Product & Service Notifications, including emails about org migrations, from Salesforce.
- User Email is unencrypted when Lightning Sync or Einstein Activity Capture are enabled. Lightning Sync and Einstein Activity Capture duplicate the User Email field in the database when users are added to sync configurations for those products. Even if you encrypt the User Email field with Shield Platform Encryption, this duplicate field stores user emails in the Salesforce database in an unencrypted state. For more information, see Considerations for Syncing Contacts, Considerations for Syncing Events, and Considerations for Setting Up Einstein Activity Capture.
- Event functionality that relies on user emails, especially calendar invitations, can be interrupted. Before encrypting the User Email field in production environments, Salesforce recommends that you test Activity features in a sandbox.
- You can't sort records in list views by fields that contain encrypted data. If you encrypt User email, you can't add it as a filter in reports.
- Login Discovery Handler lookups that rely on emails don't work if the email field is encrypted, which can block user logins. If your lookups rely on emails, don't encrypt the User Email field.
- If you use Einstein Conversation Insights, encrypt User Email with case-insensitive deterministic encryption. Some Einstein Conversation Insights features, including video calls, don't work when User Email is encrypted with probabilistic encryption.

Flows and Processes

You can reference encrypted fields in most places in your flows and processes. However, you can't reference encrypted fields in these filtering or sorting contexts.

Tool	Filtering Availability	Sorting Availability
Process Builder	Update Records action	n/a
Flow Builder	Record Choice Set resource Get Records element Delete Records element Update Records element	Record Choice Set resource Get Records element

Tool	Filtering Availability	Sorting Availability
	Condition requirements	

You can store the value from an encrypted field in a variable and operate on that value in your flow's logic. You can also update the value for an encrypted field.

Paused flow interviews can cause data to be saved in an unencrypted state. When a flow or process is waiting to resume, the associated flow interview is serialized and saved to the database. The flow interview is serialized and saved when:

- Users pause a flow
- Flows execute a Wait element
- Processes are waiting to execute scheduled actions

If the flow or process loads encrypted fields into a variable during these processes, that data isn't always encrypted at rest.

Next Best Action Recommendations

When you use probabilistic encryption, you can't use encrypted fields like Recommendation Description when you specify conditions to load recommendations.

Custom Fields

You can't use encrypted custom fields in criteria-based sharing rules.

Some custom fields can't be encrypted.

- Fields that have the `Unique` or `External ID` attributes or include these attributes on previously encrypted custom fields (applies only to fields that use the probabilistic encryption scheme)
- Fields on external data objects
- Fields that are used in an account contact relation

You can't use Schema Builder to create an encrypted custom field.

You can't use Shield Platform Encryption with Custom Metadata Types.

Masking Tradeoffs

Shield Platform Encryption doesn't provide a masking feature, but it encrypts fields that you configure with masking. We reserve a few values to notify you when the encryption key used for an encrypted masked field is unavailable or has been destroyed. The topic Why Isn't My Encrypted Data Masked? on page 32 lists all the reserved masking notification strings.

SOQL and SOSL

- You can't include fields encrypted with the probabilistic encryption scheme in the following SOQL and SOSL clauses and functions:
  - Aggregate functions such as MAX(), MIN(), and COUNT_DISTINCT()
  - WHERE clause
  - GROUP BY clause
  - ORDER BY clause

For information about SOQL and SOSL compatibility with deterministic encryption, see Considerations for Using Deterministic Encryption in Salesforce Help.

Tip: Consider whether you can replace a WHERE clause in a SOQL query with a FIND query in SOSL.

- When you query encrypted data, invalid strings return an `INVALID_FIELD` error instead of the expected `MALFORMED_QUERY`.

Marketing Cloud Account Engagement

Account Engagement supports contact email addresses encrypted by Shield Platform Encryption as long as your instance meets a few conditions. Your org must allow multiple prospects with the same email address. After this feature is enabled, you can add the contact email address field to your encryption policy.

Because the contact email address shows in the Permission object, users must have permission to view the Prospect object.

If you encrypt the contact email address field, the Salesforce Connector can't use the email address as a secondary prospect match criteria. For more information, read Salesforce Connector Settings.

Portals

If a legacy portal (created before 2013) is enabled in your org, you can't encrypt standard fields. To enable encryption on standard fields, deactivate all legacy customer and partner portals. (Salesforce Experience Cloud sites are supported.)

To deactivate a legacy customer portal, go to the Customer Portal Settings page in Setup. To deactivate a legacy partner portal, go to the Partners page in Setup.

Salesforce B2B Commerce

Shield Platform Encryption supports version 4.10 and later of the Salesforce B2B Commerce managed package, with some behavior differences. For a complete list of considerations, see Enable Shield Platform Encryption for B2B Commerce for Visualforce Objects.

Search

If you encrypt fields with a key and then destroy the key, the corresponding search terms remain in the search index. However, you can't decrypt the data associated with the destroyed key.

Accounts, Person Accounts, and Contacts

When Person Accounts are turned on, encrypting any of the following Account fields encrypts the equivalent Contact fields, and vice versa.

- Name
- Description
- Phone
- Fax

When you encrypt any of the following Account or Contact fields, the equivalent fields in Person Accounts are also encrypted.

- Name
- Description
- Mailing Address
- Phone

- Fax
- Mobile
- Home Phone
- Other Phone
- Email

When the Account Name or Contact Name field is encrypted with probabilistic encryption, searching for duplicate accounts or contacts to merge doesn't return any results. With deterministic encryption, searching for duplicate accounts or contacts to merge will find duplicates.

When you encrypt the First Name or Last Name field on a contact, that contact appears in the Calendar Invite lookup only if you haven't filtered by First Name or Last Name.

Data copied from an encrypted Contact field to a Quote field isn't encrypted.

Email Bounce Handling

Bounce handling doesn't support encrypted email addresses. If you need email bounce handling, don't encrypt the standard Email field.

Email-to-Case

Copying text from email fields also copies unicode characters embedded in email text. Two of those unicode character sequences, `\uFFFE` and `\uFFFF`, can't be included in text encrypted by Shield Platform Encryption. If you encounter an error mentioning these unicode sequences, delete the text copied from the email field and type it manually.

Activity Subject and Description

You can encrypt an Activity Subject field with case-insensitive encryption. If you destroy key material that encrypts a field, filtering on the field doesn't yield matches.

If you encrypt the Activity Subject field and it's used in a custom picklist, delete and replace actions aren't available for that value. To remove an Activity Subject value from a picklist, deactivate it.

Activity Subject fields that include an OrgID aren't copied over when you create a sandbox copy of a production org.

Encrypting Activity Description also encrypts the Task Comment field. The validation email lists the Task Comment field but not Activity Description, even though both fields are encrypted.

Salesforce for Outlook

If you encrypt the same fields that you filter in Salesforce for Outlook datasets, Salesforce for Outlook doesn't sync. To get Salesforce for Outlook to sync again, remove the encrypted fields from your filters in your datasets.

Campaigns

Campaign member search isn't supported when you search by encrypted fields.

Notes

You can encrypt the body text of Notes created with the new Notes tool. However, the Preview file and Notes created with the old Notes tool aren't supported.

Field Audit Trail

Data in a previously archived Field Audit Trail isn't encrypted when you turn on Platform Encryption. For example, say that your org uses Field Audit Trail to define a data history retention policy for an account field, such as the phone number field. When you turn on encryption for that field, new phone number records are encrypted as they're created. Previous updates to the phone number field that are stored in the Account History related list are also encrypted. However, phone number history data that is already archived in the `FieldHistoryArchive` object is stored without encryption. To encrypt previously archived data, contact Salesforce.

Salesforce Experiences

If you encrypt the Account Name field and you're not using Person Accounts, encryption affects how users' roles are displayed to admins. Normally, a site user's role name is displayed as a combination of their account name and the name of their user profile. When you encrypt the Account Name field, the account ID is displayed instead of the account name.

For example, when the Account Name field isn't encrypted, users belonging to the Acme account with the Customer User profile would have a role called `Acme Customer User`. When Account Name is encrypted (and Person Accounts aren't in use), the role is displayed as something like `001D000000IRt53 Customer User`.

Data Import Wizard

You can't use the Data Import Wizard to perform matching using master-detail relationships or update records that contain fields that use the probabilistic encryption scheme. You can use it to add new records, however.

Reports, Dashboards, and List Views

- Report charts and dashboard components that display encrypted field values might be cached unencrypted.
- You can't sort records in list views by fields that contain encrypted data.

Encryption for Chatter

When you embed a custom component in your Chatter feed using Rich Publisher Add-Ons, the data related to those add-ons is encoded, but it isn't encrypted with the Shield Platform Encryption service. Unencrypted data in Rich Publisher Add-Ons includes data stored in the Extension ID, Text Representation, Thumbnail URL, Title, Payload, and PayloadVersion fields.

Encryption for Custom Matching Rules Used in Duplicate Management

Custom matching rules can only reference fields encrypted with the deterministic encryption scheme. Probabilistic encryption isn't supported. When you rotate your keys, you must deactivate and then reactivate custom matching rules that reference encrypted fields. If you don't take this step after updating your key material, matching rules don't find all your encrypted data.

Standard matching rules that include fields with Shield Platform Encryption don't detect duplicates. If you encrypt a field included in standard matching rules, deactivate the standard rule.

Service protections ensure that loads are balanced across the system. The matching service searches for match candidates until it finds all matches up to 200 matches. With Shield Platform Encryption, the service search maximum is 100 candidates. With encryption, you could find fewer or no possible duplicate records.

Duplicate jobs aren't supported.

Self-Service Background Encryption

Self-service background encryption can encrypt data once every 7 days. This limit includes synchronization processes initiated from the Encryption Statistics and Data Sync page, synchronization that automatically runs when you disable encryption on a field, and synchronization completed by Salesforce Customer Support at your request.

Some conditions prevent the self-service background encryption from running:

- There are more than 10 million records in an object
- The org has destroyed key material
- An object's data is already synchronized
- The synchronization process is already running, initiated either by the customer or by Salesforce Customer Support at the customer's request
- Statistics are being gathered
- An encryption policy change is being processed, such as enabling encryption on a field or data element

After you begin the synchronization process, wait until it finishes before changing your encryption policy or generating, uploading, or deleting key material. These actions abort the synchronization process.

Employees

If the email field is encrypted using probabilistic encryption, wellness check surveys can't be used. Deterministic encryption is fully supported.

Messaging End User

Encrypting fields on the Messaging End User object sometimes affects indexing. If you see performance degradation on these fields, manually create custom indexes on the affected fields after enabling encryption.

General

- Encrypted fields can't be used in:
  - Criteria-based sharing rules
  - Similar opportunities searches
  - External lookup relationships
- Fields encrypted with the probabilistic encryption scheme can't be used in filter criteria for data management tools. For considerations specific to filter-preserving deterministic encryption, read Considerations for Using Deterministic Encryption.
- Web-to-Case is supported, but the Web Company, Web Email, Web Name, and Web Phone fields aren't encrypted at rest.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

Considerations for Using Deterministic Encryption

These considerations apply to data encrypted with Shield Platform Encryption's deterministic encryption scheme. Some considerations manifest differently depending on whether data is encrypted with the case-sensitive or case-insensitive deterministic encryption scheme.

API Options to Identify Filterable Fields

Fields encrypted using the deterministic encryption scheme are filterable. You can use the `isFilterable()` method to determine the encryption scheme of a particular encrypted field. If the field is filterable, the method returns true.

However, you can't explicitly detect or set the deterministic encryption scheme via the API.

Available Fields and Other Data

Deterministic encryption is available for custom URL, email, phone, text, and text area field types. It isn't available for other types of data:

- Custom date, date/time, long text area, rich text area, or description field types
- Chatter
- Files and attachments

Case Sensitivity

When you use case-sensitive deterministic encryption, case matters. In reports, list views, and SOQL queries on encrypted fields, the results are case-sensitive. Therefore, a SOQL query against the Contact object, where LastName = Jones, returns only Jones, not jones or JONES. Similarly, when the case-sensitive deterministic scheme tests for unicity (uniqueness), each version of "Jones" is unique.

Chat

For the best possible recommendation results, use the case-sensitive deterministic encryption scheme with the Utterance field on the Utterance Suggestion object. This field doesn't support other encryption schemes at this time.

The Actor Name field on the Conversation Entry object supports case-sensitive deterministic encryption, but not case-insensitive deterministic encryption.

Compound Fields

Even with deterministic encryption, some kinds of searches don't work when data is encrypted with case-sensitive deterministic encryption. Concatenated values, such as compound names, aren't the same as the separate values. For example, the ciphertext for the compound name "William Jones" isn't the same as the concatenation of the ciphertexts for "William" and "Jones".

So, if the First Name and Last Name fields are encrypted in the Contacts object, this query doesn't work:

```
Select Id from Contact Where Name = 'William Jones'
```

But this query does work:

```
Select Id from Contact Where FirstName = 'William’ And LastName ='Jones'
```

Case-sensitive and case-insensitive deterministic encryption schemes support compound fields, but only with individual column queries.

Converting Account and Contact Records to Person Accounts

When you convert account and contact records to Person Accounts, synchronize your data. Syncing resets the indexes that allow case-insensitive filtering.

Custom Field Allocations

To allow case-insensitive queries, Salesforce stores a lowercase duplicate of your data as a custom field in the database. These duplicates are necessary to enable case-insensitive queries, but they count against your total custom field count.

External ID

Case-insensitive deterministic encryption supports Text and Email external ID custom fields but not other external ID custom fields. When you create or edit these fields, use one of the recommended field setting combinations.

External ID Field Type	Unique Attributes	Encrypted
Text	None	Use case-insensitive deterministic encryption
Text	Unique and case sensitive	Use case-sensitive deterministic encryption
Text	Unique and case insensitive	Use case-insensitive deterministic encryption
Email	None	Use case-insensitive deterministic encryption
Email	Unique	Use case-sensitive deterministic encryption

You can't save changes to both Unique - Case-Sensitive and Encrypted options at the same time. Change one setting, save it, then change the next.

Filter Operators

In reports and list views, the operators "equals" and "not equal to" are supported with case-sensitive deterministic encryption. Other operators, like "contains" or "starts with," don't return an exact match and aren't supported. Features that rely on unsupported operators, such as Refine By filters, also aren't supported.

Case-insensitive deterministic encryption supports list views and reports. However, the user interface displays all operators, including operators that aren't supported for encrypted data. To review the list of supported operators available in Salesforce Classic, see Use Encrypted Data in Formulas.

Filter Records by Strings

You can search for records using strings. However, commas in strings act as OR statements. If your string includes a comma, use quotation marks around the string. For example, a search for *"Universal Containers, Inc, Berlin"* returns records that include the full string, including the comma. Searches for *Universal Containers, Inc, Berlin* returns records that include "Universal Containers" or "Inc" or "Berlin".

Formulas

Fields encrypted with the deterministic encryption scheme can't be referenced in SOQL WHERE queries.

Indexes

Case-sensitive deterministic encryption supports single-column indexes, single-column case-sensitive unique indexes, two-column indexes, and custom indexes on standard and custom fields.

Case-insensitive deterministic encryption offers limited support for standard indexes on these standard fields.

- Contact—Email
- Email Message—Relation
- Lead—Email
- Name

Queries against these fields, when encrypted with case-insensitive deterministic encryption, can perform poorly with large tables. For optimal query performance, use custom indexes instead of standard indexes. To set up custom indexes, contact Salesforce Customer Support. Lookup fields that reference the Name field also follow this pattern because they rely on indexes. To filter on the Name field in list views and reports, filter against the standard Name field instead of a lookup field.

Expect the enablement process to take longer when you apply deterministic encryption to a field with a large number of records. To support filtering, the enablement process also rebuilds field indexes.

Key Rotation and Filter Availability

When you rotate key material or change a field's encryption scheme to case-sensitive deterministic encryption or case-insensitive deterministic encryption, synchronize your data. Syncing applies the active Fields (Deterministic) key material to existing and new data. If you don't sync your data, filtering and queries on fields with unique attributes don't return accurate results.

You can sync most data yourself from the Encryption Statistics and Data Sync page in Setup. See Synchronize Your Data Encryption with the Background Encryption Service.

Next Best Action Recommendations

When you use deterministic encryption, you can use encrypted fields in load conditions only with the equals or not equals operator.

SOQL GROUP BY Statements

You can use most of the SOQL statements with deterministic encryption. One exception is GROUP BY, which isn't supported, even though you can group report results by row or column.

SOQL LIKE and STARTS WITH Statements

Deterministic encryption only supports exact, case-sensitive matches. Comparison operators that return partial matches aren't supported. For example, LIKE and STARTS WITH statements aren't supported.

SOQL ORDER BY Statements

Because deterministic encryption doesn't maintain the sort order of encrypted data in the database, ORDER BY isn't supported.

Shield Platform Encryption and the Lightning Experience

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

Notes

Note previews in Lightning are not encrypted.

File Encryption Icon

The icon that indicates that a file is encrypted doesn’t appear in Lightning.

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

Field Limits with Shield Platform Encryption

It’s good practice to use validation rules to enforce these field limits. In addition, because encrypted content is often longer than its ciphertext, encrypting a field can impose further limits on the values that you store in that field. Therefore, test your field limits in longer fields, such as Address and Subject, and on any encrypted field that contains non-ASCII values such as Chinese, Japanese, or Korean-encoded data.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

	API Length	Byte Length	Non-ASCII Characters
Assistant Name (Contact)	40	120	22
Address (To, CC, BCC on Email Message) (when encrypted with probabilistic or case-sensitive deterministic encryption)	2959	4000	1333
City (Account, Contact, Lead)	40	120	22
Email (Contact, Lead)	80	240	70
Fax (Account)	40	120	22
First Name (Account, Contact, Lead)	40	120	22
Last Name (Contact, Lead)	80	240	70
Middle Name (Account, Contact, Lead)	40	120	22
Name (Custom Object)	80	240	70
Name (Opportunity)	120	360	110

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

	API Length	Byte Length	Non-ASCII Characters
Phone (Account, Contact)	40	120	22
Site (Account)	80	240	70
Subject (Email Message)(when encrypted with probabilistic or case-sensitive deterministic encryption)	2207	3000	1000
Title (Contact, Lead)	128	384	126

Note: This list isn't exhaustive. For information about a field not shown here, refer to the API.

Reported API Lengths of Encrypted Fields

To query the length of a field using Apex, you can use the Schema.DescribeFieldResult class, which provides metadata information about a field. The `getByteLength()` and `getLength()` methods return the original length defined for the field before encryption, not the actual length of either the encrypted data or its plaintext.

For example, suppose you have an email address field defined with a length of 99 bytes. A user stores the value `aaa@aaa.aaa`, When encrypted, the field contains `txagearxhoxcrypabef'`. These values are both shorter than 99 bytes. Querying the length of this field with `DescribeFieldResult.getByteLength()` returns 99.

Email Message Fields and Case-Insensitive Encryption

To encrypt Address and Subject fields on the Email Message object with case-insensitive deterministic encryption, apply the scheme before you enter data into these fields. If existing data in these fields exceeds the following limits, that data isn't encrypted with case-insensitive deterministic encryption.

- API length: 527
- Byte length: 765
- Non-ASCII characters: 262

Case Comment Object

The Body field on the Case Comment object has a limit of 4,000 ASCII characters (or 4,000 bytes). However, when the Body field is encrypted, the character limit is lower. How much lower depends on the kind of characters you enter.

- ASCII: 2959
- Chinese, Japanese, Korean: 1333
- Other non-ASCII: 1479

Which Salesforce Apps Don't Support Shield Platform Encryption?

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

These apps don't support data encrypted with Shield Platform Encryption.

- Connect Offline
- Commerce Cloud (Salesforce B2B Commerce version 4.10 and later is supported)
- Einstein Recommendation Engine in Marketing Cloud Engagement (includes Einstein Recommendations, Einstein Web Recommendations, and Einstein Email Recommendations)
- Salesforce Einstein (includes Einstein Search, Sales Cloud Einstein, Einstein Discovery, Einstein Builders, and Einstein Vision and Language)
- Heroku (but Heroku Connect does support encrypted data)
- Marketing Cloud (but Marketing Cloud Connect does support encrypted data)
- Sales productivity features that require data to be stored using a public cloud provider
- Social Customer Service
- Thunder
- Quip
- Salesforce Billing

Legacy portals (customer, self-service, and partner) don't support data encrypted with Shield Platform Encryption. If legacy portals are active, Shield Platform Encryption can't be enabled.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?

EDITIONS

Available in both Salesforce Classic (not available in all orgs) and Lightning Experience.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield or Shield Platform Encryption. Available in **Developer** Edition at no charge.

INDEX