<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strengthen Your Data's Security with Shield Platform Encryption</td>
<td>1</td>
</tr>
<tr>
<td>Encrypt Fields and Files</td>
<td></td>
</tr>
<tr>
<td>Encrypt Fields</td>
<td>3</td>
</tr>
<tr>
<td>Encrypt Files</td>
<td>4</td>
</tr>
<tr>
<td>Fix Blockers</td>
<td>5</td>
</tr>
<tr>
<td>Retrieve Encrypted Data with Formulas</td>
<td>6</td>
</tr>
<tr>
<td>Encrypt Data in Chatter (Pilot)</td>
<td>8</td>
</tr>
<tr>
<td>Manage Shield Platform Encryption</td>
<td>9</td>
</tr>
<tr>
<td>Generate a Secret</td>
<td>10</td>
</tr>
<tr>
<td>Rotate Keys</td>
<td>15</td>
</tr>
<tr>
<td>Export a Key</td>
<td>16</td>
</tr>
<tr>
<td>Destroy a Key</td>
<td>16</td>
</tr>
<tr>
<td>Stop Encryption</td>
<td>17</td>
</tr>
<tr>
<td>How Encryption Works</td>
<td>17</td>
</tr>
<tr>
<td>What Gets Encrypted?</td>
<td>18</td>
</tr>
<tr>
<td>Can I Bring My Own Encryption Key?</td>
<td>22</td>
</tr>
<tr>
<td>Permissions</td>
<td>26</td>
</tr>
<tr>
<td>Masked Data</td>
<td>27</td>
</tr>
<tr>
<td>Encryption Flow</td>
<td>28</td>
</tr>
<tr>
<td>Behind the Scenes: The Search Index Encryption Process</td>
<td>29</td>
</tr>
<tr>
<td>Deployment</td>
<td>30</td>
</tr>
<tr>
<td>Sandbox</td>
<td>31</td>
</tr>
<tr>
<td>Terminology</td>
<td>31</td>
</tr>
<tr>
<td>Classic Encryption</td>
<td>32</td>
</tr>
<tr>
<td>Encryption Best Practices</td>
<td>33</td>
</tr>
<tr>
<td>Encryption Trade-Offs</td>
<td>35</td>
</tr>
<tr>
<td>General Trade-Offs</td>
<td>36</td>
</tr>
<tr>
<td>App Trade-Offs</td>
<td>40</td>
</tr>
<tr>
<td>Lightning Trade-Offs</td>
<td>41</td>
</tr>
<tr>
<td>Field Limits</td>
<td>41</td>
</tr>
</tbody>
</table>
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. It enables you to encrypt sensitive data at rest, and 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.

Shield Platform Encryption builds on the data encryption options that Salesforce offers out of the box. Data stored in many standard and custom fields and in files and attachments is encrypted using an advanced HSM-based key derivation system, so it is protected even when other lines of defense have been compromised.

Your data encryption key is never saved or shared across organizations. Instead, it is derived on demand from a master secret and your organization-specific tenant secret, and cached on an application server.

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

IN THIS SECTION:

Encrypt Fields and Files
Specify the fields and files you want to encrypt. Remember that encryption is not the same thing as field-level security or object-level security. Those should already be in place before you implement your encryption strategy.

Manage Shield Platform Encryption
To provide Shield Platform Encryption for your organization, contact your Salesforce account executive. They’ll help you provision the correct license so you can get started on creating your own unique tenant secret.

How Shield Platform Encryption Works
Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that’s maintained by Salesforce. We combine these secrets to create your unique data encryption key. We use that key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

Platform Encryption Best Practices
Take the time to identify the most likely threats to your organization. This will help 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.

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.
Encrypt Fields and Files

Specify the fields and files you want to encrypt. Remember that encryption is not the same thing as field-level security or object-level security. Those should already be in place before you implement your encryption strategy.

IN THIS SECTION:

Encrypt Fields
Select the fields you want to encrypt. For best results, encrypt the smallest possible number of fields.

Encrypt 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.

Fix Compatibility Problems
When you select fields or files to encrypt, Salesforce automatically checks for potential side effects and 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.

Retrieve Encrypted Data with Formulas
Use custom formula fields to quickly find encrypted data. You can write formulas with several operators and functions, render encrypted data in text, date, and date/time formats, and reference quick actions.

Encrypt Data in Chatter (Pilot)
Enabling Shield Platform Encryption for Chatter adds an extra layer of security to information that users share in Chatter. You can encrypt data at rest in feed posts, questions and answers, link names and URLs, comments, and poll questions and choices.

EDITIONS


Available in both Salesforce Classic and Lightning Experience.
Encrypt Fields

Select the fields you want to encrypt. For best results, encrypt the smallest possible number of fields.

Depending on the size of your organization, enabling a standard field for encryption can take a few minutes.

1. Make sure that your organization has an active encryption key. If you’re not sure, check with your administrator.
2. From Setup, use the Quick Find box to find the Platform Encryption setup page.
3. Click Encrypt Fields.
4. Click Edit.
5. Select the fields you want to encrypt, and save your settings.

The automatic Platform Encryption validation service will now check for settings in your organization that might block encryption. You’ll receive an email with suggestions for fixing any incompatible settings.

Field values are automatically encrypted only in records created or updated after you’ve enabled encryption. Salesforce recommends updating existing records to ensure that their field values are encrypted. For example, if you encrypt the Description field on the Case object, use the Data Loader to update all case records. Contact Salesforce if you need help with this.

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

SEE ALSO:
- Which Fields Can I Encrypt?
- Field Limits with Shield Platform Encryption
- Data Loader
- Fix Compatibility Problems
- Encrypt Files and Attachments
Encrypt 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 organization has an active encryption key; if you’re not sure, check with your administrator.

1. From Setup, enter Platform Encryption in the Quick Find box, then select Platform Encryption.
2. Select Encrypt Files and Attachments.
3. Click Save.

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 were already in Salesforce. To encrypt existing files, contact Salesforce.

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).

Here’s What It Looks Like When a File Is Encrypted.

SEE ALSO:
Encrypt Fields
Fix Compatibility Problems

When you select fields or files to encrypt, Salesforce automatically checks for potential side effects and 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 customer portal or a partner portal is enabled in your organization. To deactivate a customer portal, go to the Customer Portal Settings page in Setup. To deactivate a partner portal, go to the Partners page in Setup.

*Note:* Communities are not related to this issue. They are 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, and NULLVALUE, as well as concatenation (&).

**Skinny tables**
You’ve selected a field that’s used in a skinny table.

**Flows and Processes (Beta)**
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 dynamic record choice
- To sort data in a dynamic record choice

*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?

**SEE ALSO:**
Encrypt Fields
Retrieve Encrypted Data with Formulas

Use custom formula fields to quickly find encrypted data. You can write formulas with several operators and functions, render encrypted data in text, date, and date/time formats, and reference quick actions.

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 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 is 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 need to 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 are encrypting fields that have multiple formula fields pointing to them, encrypt those fields one at a time.

**Important:** Beginning in Spring '17, Shield Platform Encryption no longer masks encrypted data. To get the most out of encryption support for custom formula field types, we recommend that you approve the “Turn Off Masking for Encrypted Data” critical update.

To activate this critical update:
1. Review your field-level security settings for any field types that include encrypted data. Ensure that field access is properly set in your org.
2. From Setup, enter Critical Updates in the Quick Find box and select Critical Updates.
3. For Turn Off Masking for Encrypted Data, click Activate.
4. Refresh your browser page.

## Encrypt Data in Chatter (Pilot)

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

**Note:** We provide encryption for Chatter to selected customers through a pilot program that requires agreement to specific terms and conditions. To be nominated to participate in the program, contact Salesforce. Pilot programs are subject to change, and we can’t guarantee acceptance. Encryption for Chatter isn’t generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can’t guarantee general availability within any particular timeframe or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for encryption for Chatter in the Chatter Product group in the Success Community.

We recommend that you implement encryption for Chatter in a dedicated Sandbox environment.

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

1. To enable access to this feature, first contact Salesforce.
2. Assign all users in your org the View Encrypted Data permission.
3. Make sure that your org has an active encryption key. If you’re not sure, check with your administrator.
4. From Setup, use the Quick Find box to find the Platform Encryption setup page.
5. Click Encrypt Chatter.

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

After you activate encryption for Chatter, new data that you enter into Chatter gets encrypted. Existing data is not encrypted. Mass-encryption for historic Chatter data isn’t available. To encrypt existing data, simply edit or update the data in any supported field.

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 for this pilot, deselect Encrypt Chatter and then reselect Encrypt Chatter.

**Note:** Beginning with Spring ’17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.

---

**Manage Shield Platform Encryption**

To provide Shield Platform Encryption for your organization, contact your Salesforce account executive. They’ll help you provision the correct license so you can get started on creating your own unique tenant secret.

Assign the Manage Encryption Keys, Manage Certificates, and Customize Application permissions to people you trust to manage tenant secrets and certificates. Users with the Manage Encryption Keys permission can generate, export, import, and destroy organization-specific keys. It’s a good idea to monitor the key management activities of these users regularly with the setup audit trail.

Users with both Manage Certificates and Manage Encryption Keys permissions can manage certificates and tenant secrets with the Shield Platform Encryption Bring Your Own Key (BYOK) service. You can also monitor these users’ key and certificate management activities with the setup audit trail.

Authorized developers can generate, rotate, export, destroy, and reimport tenant secrets by coding a call to the TenantSecret object in the Salesforce API.

---

**IN THIS SECTION:**

- **Generate a Tenant Secret**
  You can have Salesforce generate a unique tenant secret for your organization, or you can generate your own tenant secret using your own external resources. In either case, you manage your own tenant secret: you can rotate it, archive it, and designate other users to share responsibility for it.

- **Rotate Your Encryption Tenant Secrets**
  You control the life cycle of your data encryption keys by controlling the life cycle of your tenant secrets. It’s recommended to regularly generate a new tenant secret and archive the previously active one.

- **Back Up Your Tenant Secret**
  Your tenant secret is unique to your organization and to the specific data to which it applies. Salesforce recommends that you export your secret to ensure continued data access in cases where you need to gain access to the related data again.

- **Destroy A Tenant Secret**
  Only destroy tenant secrets in extreme cases where access to related data is no longer needed. Your tenant secret is unique to your organization and to the specific data to which it applies. Once you destroy a tenant secret, related data is not accessible unless you previously exported the key and then import the key back into Salesforce.

---

**EDITIONS**


Available in both Salesforce Classic and Lightning Experience.

**USER PERMISSIONS**

To manage tenant secrets:
- Manage Encryption Keys
Turn Shield Platform Encryption Off
At some point, you may need to 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.

SEE ALSO:
Which User Permissions Does Shield Platform Encryption Require?
The TenantSecret Object

Generate a Tenant Secret
You can have Salesforce generate a unique tenant secret for your organization, or you can generate your own tenant secret using your own external resources. In either case, you manage your own tenant secret: you can rotate it, archive it, and designate other users to share responsibility for it.

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, we strongly recommend re-encrypting these fields using the latest key. Contact Salesforce for help with this.

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

IN THIS SECTION:
Generate a Tenant Secret with Salesforce
Salesforce makes it easy to generate a unique tenant secret from the Setup menu.

Manage Tenant Secrets by Type
Tenant secret types allow you to specify which kind of data you want to encrypt with a tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files or other data stored in Salesforce.

Generate Your Own Tenant Secret (BYOK)
When you supply your own tenant secret, you get the benefits of built-in Salesforce Shield Platform Encryption plus the extra assurance that comes from exclusively managing your tenant secret.

SEE ALSO:
Permission Sets
Profiles
Generate a Tenant Secret with Salesforce

Salesforce makes it easy to generate a unique tenant secret from the Setup menu. Only authorized users can generate tenant secrets from the Platform Encryption page. Ask your Salesforce admin to assign you the Manage Encryption Keys permission.

1. From Setup, enter Platform Encryption in the Quick Find box and select Platform Encryption.
2. In the Choose Tenant Secret Type drop-down list, choose a data 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 Data in Salesforce 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.

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

Manage Tenant Secrets by Type

Tenant secret types allow you to specify which kind of data you want to encrypt with a tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files or other data stored in Salesforce.

Tenant secrets are categorized according to the kind of data they encrypt.
- Data in Salesforce, which includes fields, attachments, and files other than search index files
- Search index files

Note: Tenant secrets that were generated or uploaded before the Spring ’17 release are categorized as the Data in Salesforce type.

1. From Setup, enter Platform Encryption in the Quick Find box and select Platform Encryption.
2. In the Choose Tenant Secret Type drop-down list, choose a data type.

The Key Management section displays all tenant secrets of that data type. If you generate or upload a tenant secret while viewing tenant secrets of a particular type, it becomes the active tenant secret for that data.

To enable search index encryption, contact your Salesforce account executive or open a support ticket.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What’s the difference?
Generate Your Own Tenant Secret (BYOK)

When you supply your own tenant secret, you get the benefits of built-in Salesforce Shield Platform Encryption plus the extra assurance that comes from exclusively managing your tenant secret.

Controlling your own tenant secret entails generating a BYOK-compatible certificate, using that certificate to encrypt and secure your self-generated tenant secret, then granting the Salesforce Shield Platform Encryption key management machinery access to your tenant secret.

IN THIS SECTION:
1. **Generate a BYOK-Compatible Certificate**
   Use Salesforce to generate a certificate to encrypt the tenant secret that we’ll use to derive your org-specific data encryption key. You can generate a self-signed or certificate-authority (CA) signed certificate.

2. **Generate and Wrap Your Tenant Secret**
   Generate a random number as your tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the certificate you generated.

3. **Upload Your Tenant Secret**
   Once you have your tenant secret, upload it to Salesforce so that the Shield Platform Encryption key management machinery can use it to derive your org-specific data encryption key.

Generate a BYOK-Compatible Certificate

Use Salesforce to generate a certificate to encrypt the tenant secret that we’ll use to derive your org-specific data encryption key. You can generate a self-signed or certificate-authority (CA) signed certificate.

To create a self-signed certificate:
1. In Setup, use the Quick Find box to go to the Platform Encryption page.
2. Click **Upload Tenant Secret**.
3. Click **Create Self-Signed Certificate**.
4. Enter a unique name for your certificate in the Label field. The Unique Name field will automatically assign a name based on what you entered in the Label field.

   The **Exportable Private Key**, **Use Platform Encryption**, and **Key Size** settings are pre-selected. This ensures that your self-signed certificate is compatible with Salesforce Shield Platform Encryption.

   **Important:** You can also create a BYOK-compatible self-signed certificate from the Certificate and Key Management page. If you chose this option, you must 1) disable **Exportable Private Key**, 2) specify a 4096-bit certificate size, and 3) enable **Platform Encryption**.
5. When the Certificate and Key Detail page appears, click Download Certificate.

If you’re not sure whether a self-signed or CA-signed certificate is right for you, consult your organization’s security policy. See Certificates and Keys in Salesforce Help for more about what each option implies.

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

**Generate and Wrap Your Tenant Secret**

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

1. Generate a 256-bit tenant secret using the method of your choice.
   You can generate your tenant secret in one of two ways:
   - Use your own on-premise 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 24 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. Specify the OAEP padding scheme. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64.

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.
Upload Your Tenant Secret

Once you have your tenant secret, upload it to Salesforce so that the Shield Platform Encryption key management machinery can use it to derive your org-specific data encryption key.

1. In Setup, use the Quick Find box to go to the Platform Encryption setup page.
2. Click Upload Tenant Secret.
3. In the Upload Tenant Secret section, attach both the encrypted tenant secret and the hashed plaintext tenant secret. Click Upload.

This tenant secret automatically becomes the active tenant secret.

**Note:** The tenant secret whose certificate has the latest expiration date automatically becomes the active tenant secret.

Your tenant secret is now ready to be used for key derivation. From here on, the Salesforce key derivation server will use the tenant secret you generated to derive the org-specific key that the app server will use to encrypt and decrypt your users’ data.

4. Export your tenant secret and back it up as prescribed in your organization’s security policy.

   You’ll have to reimport the secret if you need to restore it. The exported secret is different from the key you uploaded. It is encrypted with a different key and has additional metadata embedded in it. See Back Up Your Tenant Secret in the Salesforce Help.

**Note:** This page is about Shield Platform Encryption, not Classic Encryption. What’s the difference?
Rotate Your Encryption Tenant Secrets

You control the life cycle of your data encryption keys by controlling the life cycle of your tenant secrets. It’s recommended to regularly generate a new tenant secret and archive the previously active one.

Consult your organization’s security policies to decide how often to rotate your tenant secrets. You can rotate a tenant secret once every 24 hours in production orgs and every 4 hours in sandbox environments.

The key derivation function uses a master secret, which is rotated with each major Salesforce release. Master secret rotation doesn’t impact your encryption keys or your encrypted data until you rotate your tenant secret.

1. From Setup, enter Platform Encryption in the Quick Find box, then click Platform Encryption.
2. From the Choose Tenant Secret Type dropdown, choose a data type.
3. Check the status of the data type’s tenant secrets. Existing tenant secrets are listed as active, archived, or destroyed.

   **ACTIVE**
   
   Can be used to encrypt and decrypt new or existing data.

   **ARCHIVED**
   
   Can’t encrypt new data. Can be used to decrypt data previously encrypted with this key when it was active.

   **DESTROYED**
   
   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.

4. Click Generate New Tenant Secret or Upload Tenant Secret. If uploading a customer-supplied tenant secret, upload your encrypted tenant secret and tenant secret hash.
5. If you want to re-encrypt field values with a newly generated tenant secret, contact Salesforce support.

   To update your data, export the objects via the API or run a report that includes the record ID. These actions trigger the encryption service to encrypt the existing data again using the newest key.

Note: This page is about Shield Platform Encryption, not Classic Encryption. What’s the difference?
Back Up Your Tenant Secret

Your tenant secret is unique to your organization and to the specific data to which it applies. Salesforce recommends that you export your secret to ensure continued data access in cases where you need to gain access to the related data again.

1. In Setup, use the Quick Find box to find the Platform Encryption setup page.
2. In the table that lists your keys, find the tenant secret you want and 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 number>.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 in case you need to import it back into your organization.
   Note: Your exported tenant secret is itself encrypted.
5. To import your tenant secret again, 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?

Destroy A Tenant Secret

Only destroy tenant secrets in extreme cases where access to related data is no longer needed. Your tenant secret is unique to your organization and to the specific data to which it applies. Once you destroy a tenant secret, related data is not accessible unless you previously exported the key and then import the key back into Salesforce.

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

1. In Setup, use the Quick Find box to find the Platform Encryption setup page.
2. In the table that lists your tenant secrets, go to the row that contains the one you want to destroy and 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.

File previews and content that was already cached in the user’s browser may still be visible in cleartext after you destroy the key that encrypted that content, until the user logs in again.

If you create a sandbox organization from your production organization and then destroy the tenant secret in your sandbox organization, the tenant secret still exists in the production organization.

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

At some point, you may need to 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, encrypted data is not mass-decrypted and any functionality that is affected by encryption is not restored. Contact Salesforce after disabling Platform Encryption for help finalizing your changes.

1. From Setup, use the Quick Find box to find Platform Encryption.
2. Click Encrypt Fields, then click Edit.
3. Deselect the fields you want to stop encrypting, then click Save.
   Users can see data in these fields.
4. To disable encryption for files, deselect Encrypt Files and Attachments and click Save.

The limitations and special behaviors that apply to encrypted fields persist after encryption is disabled. Any previously encrypted files and attachments may remain encrypted at rest.

Encrypted fields remain accessible after you disable encryption, as long as the key used to encrypt them has not been destroyed.

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that’s maintained by Salesforce. We combine these secrets to create your unique data encryption key. We use that key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

Encrypting files, fields, and attachments has no effect on your organization’s storage limits.

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

IN THIS SECTION:

What Gets Encrypted?
Shield Platform Encryption allows you to encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, as well as Salesforce search indexes.

Can I Bring My Own Encryption Key?
Yes. You can generate and store your tenant secret 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.
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 tenant secrets. You can enable these permissions for user profiles just like you would any other user permission.

Why Isn't All of My Encrypted Data Masked?
Shield Platform Encryption uses masking only in specific circumstances. Specifically, masked characters are used to help you troubleshoot encryption key issues. If the encryption service isn't available, data in specific encrypted fields are also masked. If you have data that you don't want users to see, revisit those users' field-level security settings, record access settings, and object permissions.

Behind the Scenes: The Shield Platform Encryption Process
When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The encryption service then encrypts the data on the application server.

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 to allow Salesforce to scale operations. Apache Lucene is used for its core library.

How Do I Deploy Shield Platform Encryption?
When you deploy Shield Platform Encryption to your organization with a tool such as Force.com IDE, Migration Tool, or Workbench, the Encrypted field attribute persists. However, if you deploy to organizations with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target organization.

How Does Shield Platform Encryption Work In a Sandbox?
Refreshing a sandbox from a production organization creates an exact copy of the production organization. If Shield Platform Encryption is enabled on the production organization, all encryption settings are copied, including tenant secrets created in production.

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 the key terms, such as hardware security module, key rotation, and master secret.

What's the Difference Between Classic Encryption and Shield Platform Encryption?
With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

What Gets Encrypted?
Shield Platform Encryption allows you to encrypt a wide variety of standard fields and custom fields. You can also encrypt files and attachments stored in Salesforce, as well as Salesforce search indexes.

IN THIS SECTION:

Which Standard Fields Can Be Encrypted?
You can encrypt certain fields on standard objects, on custom objects, and in Chatter. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

Which Custom Fields Can Be Encrypted?
You can encrypt the contents of fields that belong to one these custom field types, on either standard or custom objects.
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.

Which Standard Fields Can Be Encrypted?
You can encrypt certain fields on standard objects, on custom objects, and in Chatter. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

Note: Beginning with Spring ’17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.

When you encrypt a field, existing values aren’t encrypted immediately. Values are encrypted only after they are touched. Contact Salesforce for help encrypting existing data.

Encrypted Standard Fields
You can encrypt the contents of these standard field types on the Account, Contact, Case, Case Comment, and Lead objects.

Accounts (Business)
- Account Name
- Billing
- Description
- Fax
- Website
- Phone
- Shipping (encrypts Street and City)
- Site

Person Accounts
- Name (Encrypts First Name, Middle Name, and Last Name)
- Mailing City

Contacts
- Assistant
- Description
- Email
- Fax
- Home Phone
- Mailing Address (Encrypts Mailing Street and Mailing City)
- Mobile
- Name (Encrypts First Name, Middle Name, and Last Name)
- Other Address (encrypts Street and City)
- Other Phone
- Phone
Strengthen Your Data’s Security with Shield Platform Encryption

What Gets Encrypted?

- Title

Cases
- Subject
- Description

Case Comments
- Body (including Internal Comments)

Leads (Beta)
- Address (Encrypts Street and City)
- Company
- Description
- Email
- Fax
- Mobile Phone
- Name (Encrypts First Name, Middle Name, and Last Name)
- Phone
- Other Phone
- Title
- Website

Chatter feed
- Feed Comment–Body
- Feed Item–Body
- Feed Item–Title
- Feed Revision–Value

These Chatter fields include feed posts, questions and answers, link names, comments, and poll questions. They don’t encrypt poll choices.

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

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

Which Custom Fields Can Be Encrypted?

You can encrypt the contents of fields that belong to one these custom field types, on either standard or custom objects.

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

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.

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
• Fields on external data objects
• Fields that are used in an account contact relation

On a custom object, the standard Name field can't be encrypted.

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

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?
Can I Bring My Own Encryption Key?

Yes. You can generate and store your tenant secret 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 tenant secret needs to meet these specifications:

- 256-bit size
- Encrypted with a public RSA key that is extracted from the downloaded BYOK certificate, then padded using OAEP padding
- Once it’s encrypted, it must be encoded in standard base64

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

IN THIS SECTION:

Why Bring Your Own Key?

Bring Your Own Key (BYOK) gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you’ve set up your key, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Take Good Care of Your Keys

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

Sample Script for Generating a BYOK Tenant Secret

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

Troubleshooting Bring Your Own Key

One or more of these frequently asked questions may help you troubleshoot any problems that arise.
Why Bring Your Own Key?

Bring Your Own Key (BYOK) gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you’ve set up your key, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Shield Platform Encryption enables Salesforce administrators to 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.

Data encryption keys aren’t stored in Salesforce. Instead, they’re derived on demand whenever a key is needed to encrypt or decrypt customer data, using a master secret and a tenant secret. The master secret is generated once per release for everyone by a hardware security module (HSM). The tenant secret is unique to your organization, and you control when it is generated, activated, and retired.

You can generate your tenant secrets in two ways:

- Use the Salesforce hardware security module (HSM) key management infrastructure to have your org-specific tenant secret generated for you.
- Use the infrastructure of your choice, such as an on-premise HSM, to generate and manage your tenant secret. This option is popularly known as “Bring Your Own Key,” although the element you’re really bringing is the tenant secret from which the key is derived.

Take Good Care of Your Keys

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

Back up all imported tenant secrets after you upload them to Salesforce to ensure that you have copies of your active tenant secrets. See Back Up Your Tenant Secret in the 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 that isn’t backed up, Salesforce won’t be able to help you retrieve it.
**Sample Script for Generating a BYOK Tenant Secret**

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

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 you have write permission to the file and use `chmod 775` to make it executable.
3. The script generates a number of files. Look for the two files that end with the `.b64` suffix. 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.

**Troubleshooting Bring Your Own Key**

One or more of these frequently asked questions may help you troubleshoot any problems that arise.

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

Make sure that you are running the right script for your operating system. If you are 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 is not 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 would like 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. Just make sure that the end 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 won’t be able to 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 do not 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 are not met, you won’t be able to 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.

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 sure your tenant secrets and certificates are in order.

<table>
<thead>
<tr>
<th>Possible cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your files were generated with an expired certificate.</td>
<td>Check the date on your certificate. If it has expired, you can renew your certificate or use another one.</td>
</tr>
<tr>
<td>Your certificate is not active, or is not a valid Bring Your Own Key certificate.</td>
<td>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.</td>
</tr>
<tr>
<td>You haven’t attached both the encrypted tenant secret and the hashed tenant secret.</td>
<td>Make sure that you attach both the encrypted tenant secret and hashed tenant secret. Both of these files should have a .b64 suffix.</td>
</tr>
<tr>
<td>Your tenant secret or hashed tenant secret wasn’t generated properly.</td>
<td>Several problems can cause this error. Usually, the tenant secret or hashed tenant secret wasn’t generated using the correct SSL parameters. If you are 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 are using a library other than OpenSSL, check that library’s support page for help 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.</td>
</tr>
</tbody>
</table>

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. 
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 tenant secrets. You can enable these permissions for user profiles just like you would any other user permission.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Manage Encryption Keys</th>
<th>Customize Application</th>
<th>View Setup and Configuration</th>
<th>Manage Certificates</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Platform Encryption Setup page</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Platform Encryption setup Page, excluding tenant secret and certificate management</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate, destroy, export, and import tenant secrets</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query TenantSecret object via the API</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

⚠️ **Note**: Beginning with Spring '17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.

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

SEE ALSO:

Manage Shield Platform Encryption
Why Isn’t All of My Encrypted Data Masked?

Shield Platform Encryption uses masking only in specific circumstances. Specifically, masked characters are used to help you troubleshoot encryption key issues. If the encryption service isn’t available, data in specific encrypted fields are also masked. If you have data that you don’t want users to see, revisit those users’ field-level security settings, record access settings, and object permissions.

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 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.)

The following table shows the fields that use masking. All others don’t.

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Mask</th>
<th>What It Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email, Phone, Text, Text Area, Text Area (Long), URL</td>
<td>????</td>
<td>This field is encrypted, and the encryption key has been destroyed.</td>
</tr>
<tr>
<td></td>
<td>!!!!!!</td>
<td>This service is unavailable right now. For help accessing this service, contact Salesforce.</td>
</tr>
<tr>
<td>Custom Date</td>
<td>08/08/1888</td>
<td>This field is encrypted, and the encryption key has been destroyed.</td>
</tr>
<tr>
<td></td>
<td>01/01/1777</td>
<td>This service is unavailable right now. For help accessing this service, contact Salesforce.</td>
</tr>
<tr>
<td>Custom Date/Time</td>
<td>08/08/1888 12:00 PM</td>
<td>This field is encrypted, and the encryption key has been destroyed.</td>
</tr>
</tbody>
</table>
You can’t enter these masking characters into an encrypted field. For example, if a Date field is encrypted and you enter 07/07/1777, you must enter a different value before it can be saved.

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

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn’t there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The encryption service then encrypts the data on the application server.

Salesforce securely generates the master and tenant secrets by using Hardware Security Modules (HSMs). The unique key is derived by using PBKDF2, a Key Derivation Function (KDF), with the master and tenant secrets as inputs.

Shield Platform Encryption Process Flow
1. 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.

2. If so, the encryption service checks for the matching data encryption key in cached memory.

3. The encryption service determines whether the key exists.
   a. If so, the encryption service retrieves the key.
   b. If not, the service sends a derivation request to a key derivation server and returns it to the encryption service running on the Salesforce Platform.

4. After retrieving or deriving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using 256-bit AES encryption.

5. 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 master 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 to allow Salesforce to scale operations. Apache Lucene is used for its core library.

Note: Open a support ticket to enable Search Index Encryption.

Leveraging 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.

There aren’t any changes in Setup or changes to the user interface, so the added protection is seamless and determined by the organization’s encryption policy.

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

Before the search index files are encrypted, a Salesforce security administrator must enable Search Index Encryption. Admins then set up their encryption policy to determine which data elements need to be embedded with encryption. Admins configure Shield Platform Encryption by selecting fields and files to encrypt. An org-specific HSM-derived key specifically for search index encryption is derived on-demand from the tenant secret. The key material is passed to the search engine’s cache on a secure channel.

The process when a user creates or edits records:

1. The core application determines if the search index segment should be encrypted or not based on metadata.

2. If the search index segment should be encrypted, the encryption service checks for the matching search encryption key ID in the cached memory.

3. The encryption service determines if the key exists in the cache.
   a. If the key exists in the cache, the encryption service uses the key for encryption.
   b. Otherwise, the service sends a request to the core application, which in turn sends an authenticated derivation request to a key derivation server and 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.

The process is similar 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 seamlessly.

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.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your organization with a tool such as Force.com IDE, Migration Tool, or Workbench, the Encrypted field attribute persists. However, if you deploy to organizations with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target organization.

You can use change sets to deploy Shield Platform Encryption to custom fields. Regardless of how you deploy, Salesforce automatically checks to see if the implementation violates Shield Platform Encryption guidelines.

<table>
<thead>
<tr>
<th>Source Organization</th>
<th>Target Organization</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shield Platform Encryption enabled</td>
<td>Shield Platform Encryption enabled</td>
<td>The source Encrypted field attribute indicates enablement</td>
</tr>
<tr>
<td>Shield Platform Encryption enabled</td>
<td>Shield Platform Encryption not enabled</td>
<td>The Encrypted field attribute is ignored</td>
</tr>
<tr>
<td>Shield Platform Encryption not enabled</td>
<td>Shield Platform Encryption enabled</td>
<td>The target Encrypted field attribute indicates enablement</td>
</tr>
</tbody>
</table>

Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?
How Does Shield Platform Encryption Work In a Sandbox?

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

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

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.

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

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 the key terms, such as hardware security module, key rotation, and master secret.

Data Encryption
The process of applying a cryptographic function to data that results in ciphertext. The platform encryption process uses symmetric key encryption and a 256-bit Advanced Encryption Standard (AES) algorithm using CBC mode, PKCSS padding, and a randomized, 128-bit initialization vector (IV) to encrypt field-level data and files stored on the Salesforce platform. Both data encryption and decryption occur on the application servers.

Data Encryption Keys
Shield Platform Encryption uses data encryption keys to encrypt and decrypt data. Data encryption keys are derived on a key derivation server using keying material split between a per-release master secret and an organization-specific tenant secret stored encrypted in the database as a part of your organization. The 256-bit derived keys exist in memory until evicted from the cache.

Encrypted Data at Rest
Data that is encrypted when stored on disk. Salesforce supports encryption for fields stored in the database, documents stored in Files, Content Libraries, and Attachments, and archived data.

Encryption Key Management
Refers to all aspects of key management, such as key creation, processes, and storage. Tenant secret management is performed by administrators or users who have the “Manage Encryption Keys” permission.

Hardware Security Module (HSM)
Used to provide cryptography processing as well as key management for authentication. Shield Platform Encryption uses HSMs to generate and store secret material and run the function that derives data encryption keys used by the encryption service to encrypt and decrypt data.

Initialization Vector (IV)
A random sequence used with a key to encrypt data.
Key Derivation Function (KDF)
Uses a pseudorandom number generator and input such as a password to derive keys. Shield Platform Encryption uses PBKDF2 (Password-based Key Derivation Function 2) with HMAC-SHA-256.

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

Master HSM
The master HSM consists of a USB device used to generate secure, random secrets each Salesforce release. The master HSM is “air-gapped” from Salesforce’s production network and stored securely in a bank safety deposit box.

Master Secret
Used in conjunction with the tenant secret and key derivation function to generate a derived data encryption key. The master secret is updated each release by Salesforce and encrypted using the per-release master wrapping key, which is in turn encrypted with the Key Derivation Servers’ public key so it can be stored encrypted on the file system. Only HSMs can decrypt it. No Salesforce employees have access to these keys in cleartext.

Master Wrapping Key
A symmetric key is derived and used as a master wrapping key, also known as a key wrapping key, encrypting all the per-release keys and secrets bundle.

Tenant Secret
An organization-specific secret used in conjunction with the master secret and key derivation function to generate a derived data encryption key. When an organization administrator rotates a key, a new tenant secret is generated. To access the tenant secret via the API, refer to the TenantSecret object. No Salesforce employees have access to these keys in cleartext.

What’s the Difference Between Classic Encryption and Shield Platform Encryption?

With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Classic Encryption</th>
<th>Platform Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing</td>
<td>Included in base user license</td>
<td>Additional fee applies</td>
</tr>
<tr>
<td>Encryption at Rest</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Native Solution (No Hardware or Software Required)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Encryption Algorithm</td>
<td>128-bit Advanced Encryption Standard (AES)</td>
<td>256-bit Advanced Encryption Standard (AES)</td>
</tr>
<tr>
<td>HSM-based Key Derivation</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Manage Encryption Keys Permission</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Editions

Available in both Salesforce Classic and Lightning Experience.
### Platform Encryption Best Practices

Take the time to identify the most likely threats to your organization. This will help 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.

**Note:** Beginning with Spring ’17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.

1. **Define a threat model for your organization.**

   Walk through a formal threat modeling exercise to identify the threats that are most likely to affect your organization. Use your findings to create a data classification scheme, which can help you decide what data to encrypt.

2. **Encrypt only where necessary.**

### Platform Encryption

<table>
<thead>
<tr>
<th>Feature</th>
<th>Classic Encryption</th>
<th>Platform Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate, Export, Import, and Destroy Keys</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>PCI-DSS L1 Compliance</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>Masking</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>Mask Types and Characters</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>View Encrypted Data Permission Required to Read Encrypted Field Values</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>Encrypted Standard Fields</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Encrypted Attachments, Files, and Content</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Encrypted Custom Fields</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Encrypt Existing Fields for Supported Custom Field Types</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Search (UI, Partial Search, Lookups, Certain SOSL Queries)</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>API Access</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Available in Workflow Rules and Workflow Field Updates</td>
<td></td>
<td>![ ]</td>
</tr>
<tr>
<td>Available in Approval Process Entry Criteria and Approval Step Criteria</td>
<td></td>
<td>![ ]</td>
</tr>
</tbody>
</table>

**Note:**

Beginning with Spring ’17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.
• 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.

• Before enabling encryption, fix any violations that you uncover. For example, referencing encrypted fields in a SOQL WHERE clause triggers a violation. Similarly, if you reference encrypted fields in a SOQL ORDER BY clause, a violation occurs. In both cases, fix the violation by removing references to the encrypted fields.

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 Force.com inherit Shield Platform Encryption capabilities and limitations.

6. Platform Encryption is not a user authentication or authorization tool. Use field-level security settings, page layout settings, and validation rules, not Platform Encryption, to control which users can see which data.

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. Mass-encrypt your existing data.

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, contact Salesforce.

9. Don’t use Currency and Number fields for sensitive data.

You can often keep private, sensitive, or regulated data safe without encrypting associated Currency or Number fields. Encrypting these fields could have broad functional consequences across the platform, such as disruptions to roll-up summary reports, report timeframes, and calculations, so they are not encryptable.

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.
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 this.

SEE ALSO:
Tradeoffs and Limitations of Shield Platform Encryption

### 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.

IN THIS SECTION:
- General Shield Platform Encryption Considerations
  These considerations apply to all data that you encrypt using Shield Platform Encryption.
- 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 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
  Under certain conditions, encrypting a field can impose limits on the values that you store in that field. Before deciding to encrypt a field, make sure that you know these limits.

SEE ALSO:
Platform Encryption Best Practices
General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

Note: Beginning with Spring ’17, Shield Platform Encryption no longer masks encrypted data in the presentation layer. This may affect some users’ ability to work with encrypted data. If you have data you don’t want specific users to see, revisit their field-level security settings, record access settings, and object permissions.

Leads

Lead and Case assignment rules, workflow rules, and validation rules work normally when Lead fields are encrypted. However, matching and de-duplication of records during lead import doesn’t work, and Einstein lead scoring is not available.

Apex Lead Conversion works normally, but PL-SQL-based lead conversion is not supported.

In the Lightning Email Composer, the To: field is not autopopulated when the Name or Email field is encrypted.

Note: This beta version of encryption support in Leads is production quality but has known limitations.

Flows and Processes (Beta)

Important: If your data is masked, flows and processes don’t work correctly. In Salesforce orgs created before Spring ’17, assign users who interact with flows or processes the View Encrypted Data permission. To avoid unexpected behavior, activate the Turn Off Masking for Encrypted Data critical update.

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.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Filtering Availability</th>
<th>Sorting Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Builder</td>
<td>Update Records action</td>
<td>n/a</td>
</tr>
<tr>
<td>Cloud Flow Designer</td>
<td>Dynamic Record Choice resource</td>
<td>Dynamic Record Choice resource</td>
</tr>
<tr>
<td></td>
<td>Fast Lookup element</td>
<td>Fast Lookup element</td>
</tr>
<tr>
<td></td>
<td>Record Delete element</td>
<td>Record Lookup element</td>
</tr>
<tr>
<td></td>
<td>Record Lookup element</td>
<td>Record Update element</td>
</tr>
</tbody>
</table>

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 result in data being 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 might not be encrypted at rest.

### 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
- 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.

### SOQL/SOSL

Encrypted fields can't be used with 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

**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`.

### Lightning Sync

With Shield Platform Encryption enabled, Lightning Sync syncs between users' email and calendar application and Salesforce only if the user has the "View Encrypted Data" permission.

### Lightning for Outlook

With Shield Platform Encryption enabled, Lightning for Outlook users who don't have the "View Encrypted Data" permission see masked values in Outlook for fields that are encrypted.

### Salesforce for Outlook

With Shield Platform Encryption enabled, Salesforce for Outlook syncs between Microsoft Outlook and Salesforce only if the user has the "View Encrypted Data" permission.

### Portals

If a portal is enabled in your organization, you can't encrypt standard fields. Deactivate all customer portals and partner portals to enable encryption on standard fields. (Communities are supported.)
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, searching for duplicate accounts or contacts to merge doesn’t return any results.

When you encrypt the First Name or Last Name field on a contact, that contact appears in the Calendar Inviter lookup only if you haven’t filtered by First Name or Last Name.

Salutation and Suffix field values in Contact records can appear masked to users without the “View Encrypted Data” permission, even if the field values aren’t encrypted.

Email

- When encrypted field values are included in email templates, they appear in plaintext to users with the “View Encrypted Data” permission. Otherwise, the running user’s permissions determine whether the recipient sees plaintext or masked data.
- Users without the “View Encrypted Data” permission can’t send emails using Mass Email Contacts.
- When the standard Email field is encrypted, email to Salesforce can’t receive inbound emails.
- When the standard Email field is encrypted, the detail page for Contacts, Leads, or Person Accounts doesn’t flag invalid email addresses. If you need bounce processing to work as expected, don’t encrypt the standard Email field.

Activities

Items in an Activity History related list may be displayed in plaintext even if the fields they refer to are encrypted.
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 are 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. If you need to encrypt previously archived data, contact Salesforce.

Page Layouts
If you preview a page layout as a profile without the “View Encrypted Data” permission, the preview’s sample data isn’t masked. The sample data may be blank or may appear in plaintext.

Communities
- For community users with the “View Encrypted Data” permission, data encryption doesn’t change anything about the community experience. However, 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 community 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 is not 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 001D000000IRr53 Customer User.
- Custom fields encrypted with Classic Encryption are masked for Community users even if they have the “View Encrypted Data” permission.

REST API
You don’t get autosuggestions via the REST API when a field is encrypted.

Data Import
You can’t use the Data Import Wizard to perform matching using master-detail relationships or update records that contain encrypted fields. You can use it to add new records, however.

Reports, Dashboards, and List Views
- Report charts and dashboard components that display encrypted field values may be cached unencrypted.
- You can’t sort records in list views by fields that are encrypted.
Encryption for Chatter (Pilot)

When you embed a custom component in your Chatter feed using Rich Publisher Add-Ons (Pilot), 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.

General

• Encrypted fields can’t be used in:
  – Criteria-based sharing rules
  – Similar opportunities searches
  – External lookup relationships
  – Skinny tables
  – Filter criteria for data management tools
  – Duplicate Management matching rules

• Live Agent chat transcripts are not encrypted at rest.
• Web-to-Case is supported, but the Web Company, Web Email, Web Name, and Web Phone fields are not encrypted at rest.

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

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. However, you can enable Shield Platform Encryption for other apps when these apps are in use.

• Connect Offline
• Commerce Cloud
• Data.com
• Heroku (but Heroku Connect does support encrypted data)
• Marketing Cloud (but Marketing Cloud Connect does support encrypted data)
• Pardot (but Pardot Connect supports encrypted contact email addresses if your Pardot org allows multiple prospects with the same email address)
• Salesforce Mobile Classic
• Salesforce IQ
• Social Customer Service
• Steelbrick
• Thunder
• Quip

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?
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.

Contacts and Leads
In Lightning, if either the Name or Email field is encrypted, Salesforce doesn’t suggest email recipients from contacts you’ve emailed before.

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.

Date Fields
Lightning shows 12/30/0001 as the dummy date for masking encrypted date values.

Custom Field Masking
When the encryption key is destroyed, the values of encrypted custom field values may appear in plaintext until the page is refreshed.

Field Limits with Shield Platform Encryption

Under certain conditions, encrypting a field can impose limits on the values that you store in that field. Before deciding to encrypt a field, make sure that you know these limits.

Custom Fields
If you expect users to enter non-ASCII values, such as CJK-encoded data, we recommend creating validation rules to enforce these limits:

- Email custom field type values that contain only non-ASCII characters are limited to 70 characters.
- Phone custom field type values that contain only non-ASCII characters are limited to 22 characters.

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 these fields are 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

Contact Object
When Shield Platform Encryption is enabled for the Name field on the Contact object, the character limit is lower for some character types. This is true for both Business accounts and Person accounts. ASCII character limits are not affected.

- First Name—22 non-ASCII characters
- Middle Name—22 non-ASCII characters
• Last Name—70 non-ASCII characters
• Mailing City—22 non-ASCII characters

敞开的笔记：这页是关于Shield Platform Encryption，而不是Classic Encryption。What's the difference?

SEE ALSO:
- Encrypt Fields