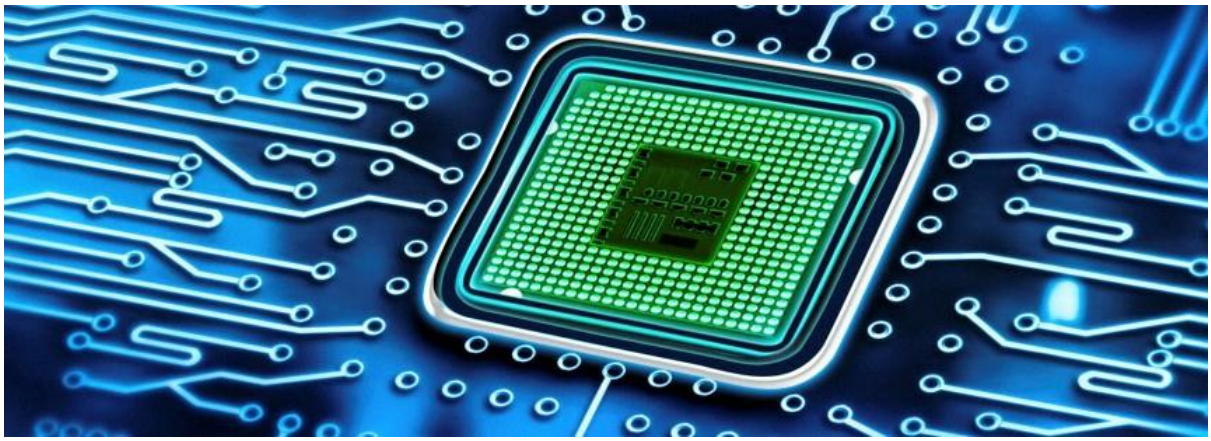
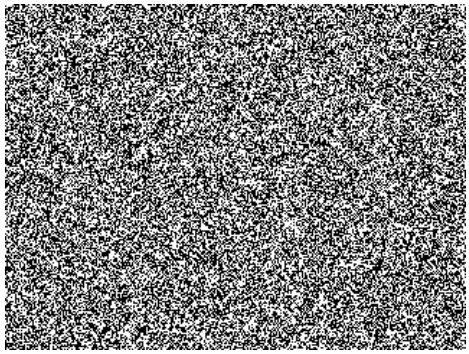
Difenso for Salesforce

Guide

I) The components of the procedure



A) Statics components:



1) Identification Service

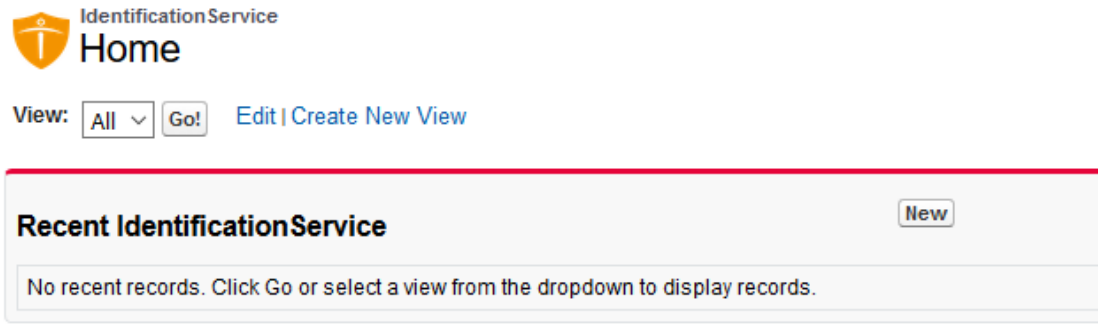
This object is a table linked to a salesforce tab. (Identification Service)
Each row of this object can contain a private key, you must create an open SSL private key and insert it.

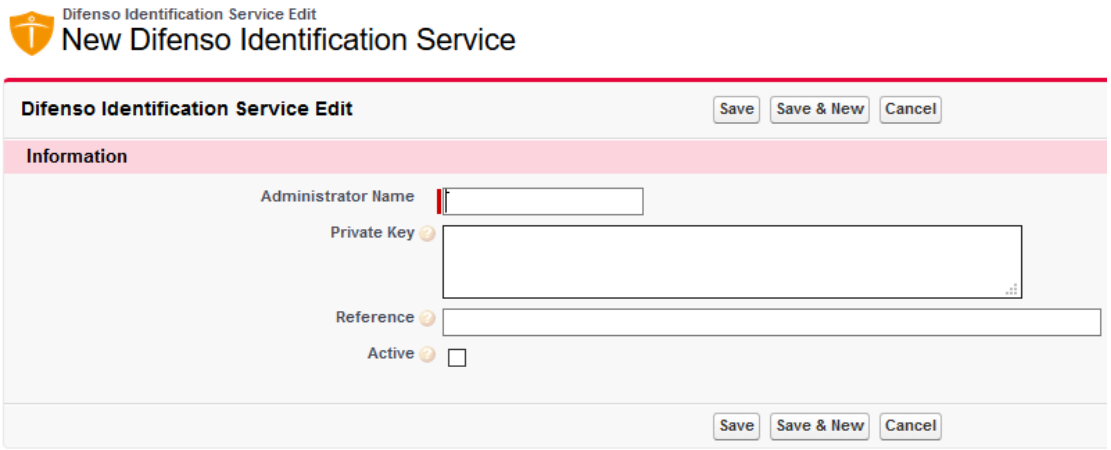
- Administrator Name: Your name

- Private key: The private Key should be in RSA's PKCS #8 (1.2) Private-Key Information Syntax Standard form. The value cannot exceed 4 KB.

- Reference of the Key

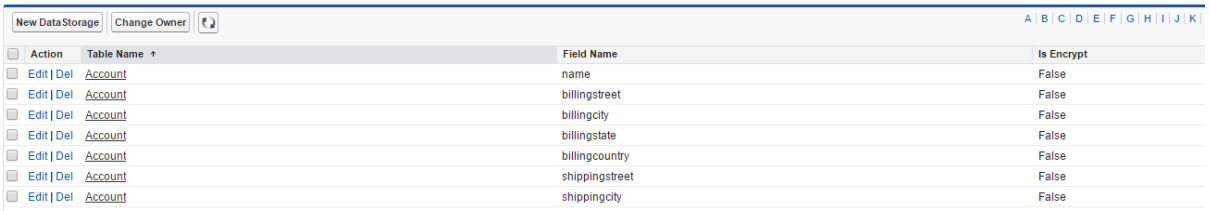
- Set the key to be Active (Only one key could be Active)





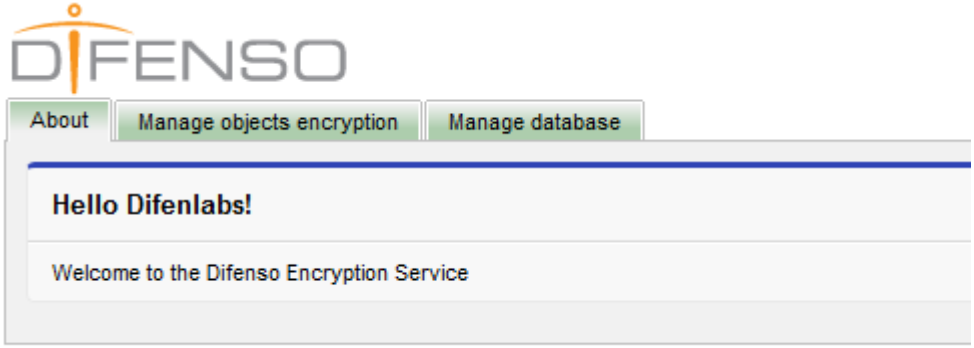
2) Data Storage

This object is a table for our service, it contains the field's name of all your objects and a Boolean True or False to know if a field should be encrypted during the process. (it is the metadata for the service)



3) Difenso Encryption Service

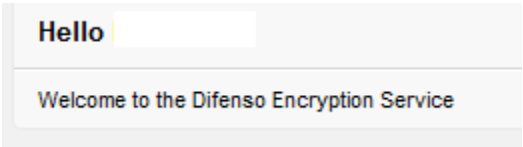
It is a visualforce page that is the space of the administrator.
In this space, you can control the encryption of the objects



There are three tabs:

a) About

This is the home page of the Difenso Encryption Service

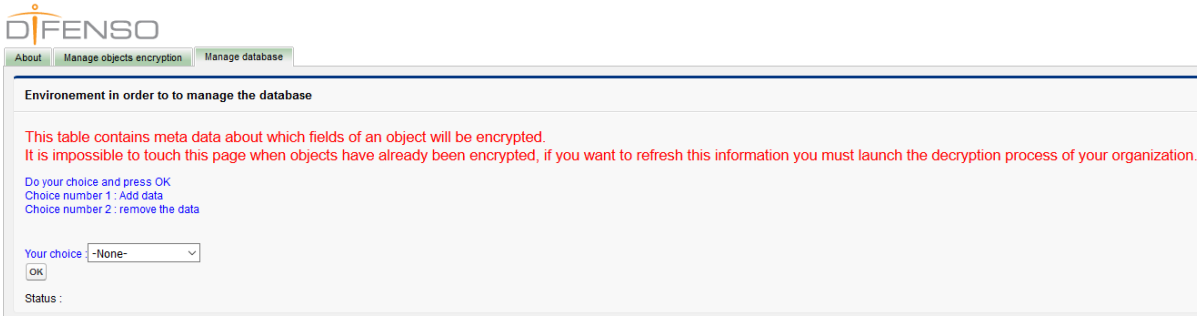


b) Manage Object Encryption

In this tab, you will choice which object and which fields should be encrypted. Before doing it, you must go to the tab Manage database.

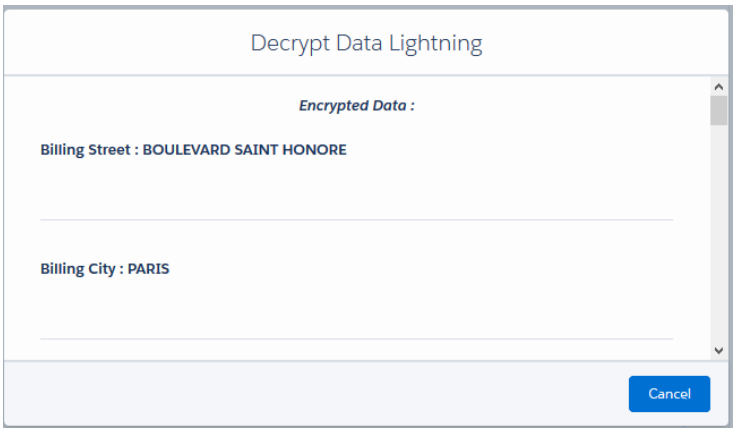
c) Manage database

In this tab, you have to add the metadata for our service (Data Storage). To begin the procedure of the encryption you have to choose the option Add data and press Ok.



4) Lightning component

This component is using by the Lightning version of Salesforce to decrypt the data.
In the Lightning version, the JavaScript button Decrypt Data doesn't work so we have created a component "Decrypt Component For Lightning"
It is a popup with a controller which display the decrypted data.



B) Dynamics components

This dynamically creation is happened in the Difenso Encryption Service in the Manage Object Encryption tab, when you clicked on save.

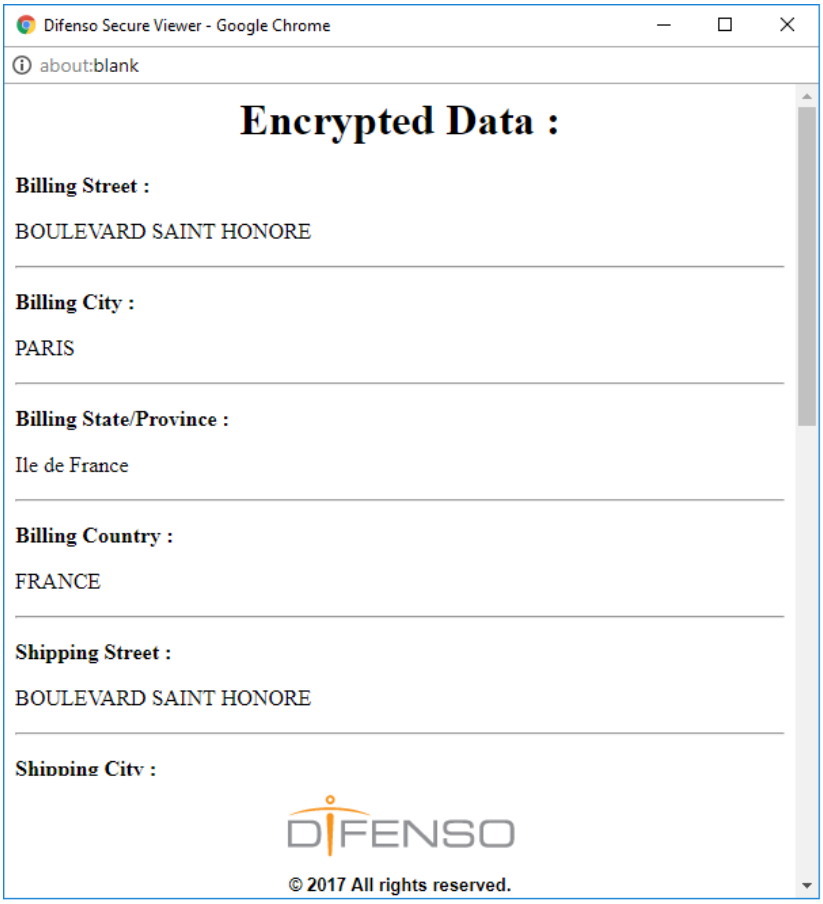
1) Trigger is created and added into your object
   This trigger will fire on every insert or every update of a record by a Salesforce user.
   This trigger will call the apex class to encrypt the data.

2) Field 'Difenso Activation' is created and added into your object
   When the service of Encryption is active, this checkbox should to be true
   It helps us to encrypt old records that were entered before the installation of the service.
   If the service is active and in a record this value is false, it means that this record was created before the installation of the service. So, this record is not yet encrypted.
   To encrypt old records, you should simply go to your record, press Update and then press Save.

3) Field 'Difenso ID_Field' is created and added into your object
   An ID is created when this trigger is fired. We use it for the process.

4) Field 'Difenso Status Processing' is created and added into your object
   When the trigger is fired, the status is changed to True.
   When the process is terminated, the status is changed to False.

5) Button 'Decrypt Data Classic' is created.
   This button when clicked, is calling the apex class to decrypt the data and display the data by opening a new page with the encrypted values.
   You should permit the use of "pop-up page" for the salesforce website.
   This button is working only on classic version.


Difenso Secure Viewer - Google Chrome
about:blank
Encrypted Data :
Billing Street :
BOULEVARD SAINT HONORE
Billing City :
PARIS
Billing State/Province :
Ile de France
Billing Country :
FRANCE
Shipping Street :
BOULEVARD SAINT HONORE
Shipping City :
DIFENSO
© 2017 All rights reserved.

II) Guide about the procedure to encrypt or decrypt data:

A) Encryption

When a salesforce user creates a new record, or updates an existing record on an encrypted object, the trigger is fired.
This trigger calls an apex class. The encryption of the data is done by two steps:

1) The first step is to prepare the data:
   a) Check if there is a valid private key
   b) Create an id from a random number for a single use.
   c) Prepare the request to the Difenso web service (signed JWT)
   d) Check which fields should be encrypted and insert in them an information word like "encrypting", "updating" or "Error"
   e) Status Processing Boolean changed to True
      This Boolean will prevent the update trigger.
   f) Save the new record and call the future function with an array of field's name, the JWT, the object name and the idKey.

2) The second step is to encrypt the data (Callout function):
   a) Catch the record to encrypt by query in the salesforce database with the random number id (idKey).
   b) Send the JWT to the Difenso web service, it will return the encrypted list fields value.
   c) Run on the fields of the object and check with our Database (Data Storage) which fields should be encrypted
   d) For each field, insert into the record query the encrypted value.
   e) Update the query.
   f) Reset the idKey (random number) and reset the Boolean Status

B) Decryption:

Classic version

In the page of the record that you want to decrypt the fields, there is a JavaScript button.
When clicking on it, an apex class is called with the id of the record.

The procedure of the JavaScript button:

1) Catch the id of the record and call the apex class with this id
2) Catch the response of the apex class
   The response is a Json
3) Convert it to a list of 2D list (Field's name, Field's Value).
4) Open a new page with the decrypted data.

The procedure of the function (in the apex class):

1) Check if there is an active private key
2) Query the record with the id
3) Check which fields should be decrypted (with the Data Storage)
4) Prepare the signed JWT with the list of Field's values and send it to the Difenso Web service
5) Split the response and save the list of Decrypted values.
6) Create a list of 2D list (Field's name, Decrypted value).
7) Convert this list of 2D list to a Json
8) Return the Json to the JavaScript button.

Lightning Experience version

When you configure your object to be encrypted, one of the steps is to create an Action Button and to link it to our Lightning component "Decrypt Component For Lightning".

So, in the page of the record that you want to decrypt the fields, there is an Action Button.
This Action Button calls the Decrypt Component For Lightning.

The procedure of the Lightning component:

1) Catch the id of the record and call the apex class with this id
2) Catch the response of the apex class
3) The response is already in apex language, so don't need to convert from Json like in the classic version
4) Display on the pop up of the Lightning component the result
5) This result is a table of rows (Field Name: Field Decrypted Value) of the encrypted value.

The procedure of the function (in the apex class):

1) Check if there is an active private key
2) Query the record with the id
3) Check which fields should be decrypted (with the Data Storage)
4) Prepare the signed JWT with the list of Field's values and send it to the Difenso Web service
5) Split the response and save the list of Decrypted values.
6) Return the list of 2D list (Field's name, Decrypted value)

