Direction des Systèmes d'Information

Procédure de déchiffrement des documents émis par Enedis sur les canaux numériques d'un client entreprise

Identification : **Enedis-NOI-CF_107E**

Version : V2

Nb. de pages : 8

Résumé / Avertissement
Le chiffrement de données transportées via Internet permet de garantir la confidentialité des informations émises par Enedis vers un client entreprise. Ces données sont émises sur les canaux de contacts définis dans la publication (adresse courriel et/ou serveur FTP). Ce document décrit :

- les modalités de chiffrement par Enedis des publications chiffrées vers les canaux numériques d'un client entreprise
- les modalités d'obtention des clés utilisées lors de ces opérations de chiffrement/déchiffrement,
- les modalités de déchiffrement, par le client entreprise, des fichiers chiffrés reçus.

Document(s) associé(s) et annexe(s) :

Version	*Date d'application*	*Nature de la modification*	*Annule et remplace*
V1	*01/06/2019*	*Création du document*	*N/A*
V2	*13/06/2023*	*Mise à jour du document et ajout d'un outil de déchiffrement*	*V1*

Accessibilité

☑ Libre ☐ *Interne* ☐ *Restreinte* ☐ *Confidentielle*

■ **Destinataires :** clients entreprise

SOMMAIRE

Procédure de déchiffrement des documents émis par Enedis sur les canaux numériques d'un client entreprise

1 — Généralités

Enedis émet des flux chiffrés pour publier des fichiers vers les clients, Collectivités et Acteurs de Marché. Les algorithmes de chiffrement utilisés par Enedis sont ceux reconnus en France comme assurant un haut niveau de confidentialité. Pour le chiffrement vers les clients entreprise, Enedis utilise l'algorithme AES (« *Advanced Encryption Standard* »), avec des clés de 256 bits.

2 — Liste des flux chiffrés à destination des clients entreprise

Le tableau suivant liste les flux chiffrés vers les clients entreprise, et indique pour chacun :

- la méthode de chiffrement utilisée par Enedis ; cette méthode pouvant évoluer dans le temps, la date d'application est précisée dans la colonne « A partir du ». Les différentes méthodes de chiffrement sont décrites dans le chapitre 4 « Les différentes méthodes de chiffrement » ;
- la méthode de gestion des clés. Les différentes méthodes de chiffrement sont décrites dans le chapitre 3 « Les différentes méthodes de gestion des clés »

Code Flux	Nom Flux	A partir du	Chiffrement	Gestion des Clés
Enedis-FOR-CF_055E	Formulaire d'habilitation aux API Enedis	Juin 2019	7-Zip / Winzip	Clé par SMS
IFJ	Infra-J	30 Juil. 2019	AES256 IV dyn	Self-Care EC Entreprise
R171	Index quotidiens	Oct. 2019	AES256 IV dyn	Self-Care EC Entreprise
R172	Relevé de glissement	Oct. 2019	AES256 IV dyn	Self-Care EC Entreprise
R4Q, R4H, R4M	Publication Récurrente de Courbe de charge	Oct. 2019	AES256 IV dyn	Self-Care EC Entreprise
R6343	Publication Récurrente de Courbe de charge	Fev. 2023	AES256 IV dyn	Self-Care EC Entreprise
R6419	Publication Récurrente de données Index	Fev. 2023	AES256 IV dyn	Self-Care EC Entreprise

3 — Les différentes méthodes de gestion des clés

A la date de création de ce document, deux méthodes de gestion des clés pour les clients entreprise sont mises en place.

3.1. Méthode « Selfcare Espace Client Entreprise (ECE) »

Sur l'Espace client entreprise, la page « Mes canaux de contact » permet à un client entreprise de gérer, en *selfcare*, ses propres canaux de contacts de type « Mail » et « FTP », à la maille d'un SIRET ou d'un SIREN.
NB : Seuls les comptes ayant étendu leur périmètre au niveau SIREN peuvent renseigner des canaux à la maille d'un SIREN.

Chaque canal de contact dispose de sa propre clé de chiffrement, qui est une clé symétrique de 256 bits qui doit rester secrète, connue seulement d'Enedis (pour chiffrer) et du client (pour déchiffrer).

Cette clé est :

— créée lors de la création d'un nouveau canal de contact dans le compte client entreprise,

— modifiée lors de la modification d'un canal de contact existant : tous les flux reçus suite à cette opération seront chiffrés avec la nouvelle clé.

Pour la lire, il faut, dans le compte client entreprise, cliquer sur l'icône « Clé » du canal de contact : elle est alors affichée sous la forme de 64 caractères hexadécimaux, et un bouton « copier » permet de placer ces 64 caractères dans le « presse-papier » pour la coller dans le système d'information chargé du déchiffrement des publications reçues sur ce canal de contact.

La clé associée à un canal de contact sera utilisée pour chiffrer tous les flux chiffrés publiés sur ce canal de contact. Autrement dit, elle n'est pas associée à un « Code flux », mais au canal lui-même.

3.2. Méthode « Clé par SMS »

Cette méthode consiste à communiquer par SMS à un interlocuteur du client entreprise la clé de déchiffrement de la pièce jointe chiffrée contenue dans un mail qui lui a été adressé en réponse à une demande ponctuelle.
Par exemple, lorsqu'un client entreprise souhaite obtenir une habilitation à l'API Enedis « Infra-J » pour utiliser la prestation F375A d'accès ponctuel aux données d'un compteur en infra-journalier, il renseigne le formulaire Enedis-FOR-CF_055E, disponible sur le site Internet, puis le transmet à Enedis, en indiquant dans ce formulaire :

— L'adresse mail de l'interlocuteur de l'entreprise chargé de recevoir la réponse d'Enedis (contenant les identifiants confidentiels)

— Le numéro de mobile sur lequel cet interlocuteur recevra le mot de passe de déchiffrement de la pièce jointe qui sera jointe au mail de réponse

Enedis créé l'habilitation et renseigne dans le même formulaire les identifiants confidentiels. Pour transmettre la réponse ainsi constituée à l'interlocuteur de l'entreprise, Enedis :

— choisit une clé de chiffrement (suite de caractères non triviale) et utilise le logiciel libre « 7-Zip File Manager » pour créer une « archive » (au sens 7-Zip, à savoir un fichier d'extension « .zip ») contenant le formulaire chiffré avec cette clé

— envoie cette archive par mail à l'interlocuteur de l'entreprise

— envoie par SMS, à l'interlocuteur, la clé de chiffrement, qui sera demandée par le logiciel « 7-Zip » ou « Winzip » pour déchiffrer l'archive.

4 — Les différentes méthodes de chiffrement / déchiffrement

Enedis utilise différents algorithmes de chiffrement.
Ce chapitre décrit, pour chaque algorithme, la méthode de chiffrement mise en œuvre par Enedis, la méthode préconisée de déchiffrement, et un exemple de code Java permettant d'implémenter le déchiffrement.
A la date de création de ce document, une seule méthode de chiffrement pour les clients entreprise est mise en place.

4.1. Méthode « AES256 IV dyn » : chiffrement AES256 avec IV dynamique

4.1.1. Méthode de chiffrement

L'algorithme AES est utilisé en mode CBC avec une clé de 256 bits et un padding « PKCS5Padding ».
Pour chaque chiffrement d'un fichier, un IV (*Initialization Vector*) de 128 bits aléatoires est créé (chaque IV ne servira qu'une seule fois, on parle donc d'IV « dynamique »). La connaissance de l'IV est indispensable pour le déchiffrement, il doit donc être transmis au destinataire. Cette transmission se fait via le fichier chiffré envoyé : les 128 bits de l'IV sont contenus dans les 128 premiers bits (ou 16 octets) du fichier.

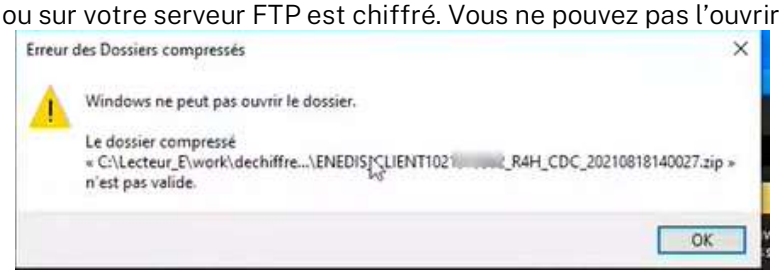
4.1.2. Méthode de déchiffrement

La méthode de déchiffrement consiste à utiliser l'algorithme AES256, en mode CBC, en utilisant l'IV lu dans les 128 premiers bits (ou 16 octets) du fichier chiffré reçu. Les données suivantes (après les 128 premiers bits) sont les blocs de données chiffrés en AES.

4.1.3. Exemple d'utilisation d'un outil de déchiffrement

Enedis met à disposition des clients, un outil de déchiffrement permettant de faciliter les manipulations réalisées lors de la réception des fichiers chiffrés. L'outil mis à disposition est présent ci-après.
NB : Merci de bien lire l'entièreté du paragraphe 4.1.3 avant de démarrer son utilisation.

Le fichier reçu par mail ou sur votre serveur FTP est chiffré. Vous ne pouvez pas l'ouvrir directement :



Etape 1 – Préparation des prérequis :
L'outil va vous permettre de déchiffrer le fichier reçu sur votre canal de contact. Il prend trois paramètres en entrée :

- La clé symétrique de chiffrement/déchiffrement de 64 bits conformément au paragraphe 3.
- Le répertoire de stockage où sont placés vos fichiers chiffrés
  A titre d'exemple, nous le plaçons dans *C:\Temp\ENEDIS*.
- Avoir JAVA sur son poste et connaître le lieu de stockage de son fichier JAVA. Téléchargeable via le lien open source, le JRE zippé x86 64-bit (première ligne) :
  https://www.openlogic.com/openjdk-downloads?field_java_parent_version_target_id=416&field_operating_system_target_id=436&field_architecture_target_id=391&field_java_package_target_id=401

Pour dézipper un fichier, il faut faire un « Clic droit » ⇨ Extraire tout ⇨ Entrer le répertoire où vous souhaitez avoir votre dossier dézippé. Le chemin d'accès de ce répertoire est à conserver pour les prochaines étapes.
A titre d'exemple, nous le plaçons dans ***C:\Users\Desktop\openlogic-openjdk-jre-8u362-b09-windows-64.***
Version JAVA pour l'exemple : ***JAVA 8, 8u362-b09, JRE, version zip***

Cette étape est primordiale et bloquera le reste de l'utilisation de l'outil.

Etape 2 – Récupération des fichiers:
Placer dans un répertoire les fichiers chiffrés et garder de côté le chemin d'accès à ce répertoire, dans notre exemple, le répertoire ENEDIS. Chemin d'accès pour l'exemple ci-dessous : *C:\Temp\ENEDIS*

Par défaut, dans l'outil de déchiffrement, le répertoire où les fichiers sont attendus est *C:\Temp\ENEDIS*. Cela peut être variabilisé à votre convenance comme nous le détaillons ci-dessous.

Etape 3 – Préparation de l'outil :
L'étape d'installation de l'outil sur votre poste est identique sous Windows ou sous Linux.

Téléchargez le fichier « dechiffrement-client » sur votre poste et placez-le dans le répertoire de votre choix. Le chemin de ce répertoire est à garder pour les prochaines étapes.
https://www.enedis.fr/media/3563/download

Le placer dans le répertoire que vous souhaitez et dézipper le fichier. Pour dézipper un fichier, il faut faire un « Clic droit » ⇨ Extraire tout ⇨ Entrer le répertoire où vous souhaitez avoir votre dossier dézippé.

Rendez-vous dans votre dossier dézippé « dechiffrement-client » et éditez votre fichier. Allez dans le dossier bin/ puis récupérez le fichier suivant pour l'éditer (Ouvrir avec un éditeur de texte type Notepad par exemple) :

dechiffrement-client.bat pour les PC Windows
dechiffrement-client.ksh pour les PC Linux

Vous arrivez ainsi sur le code de l'outil.
Les deux lignes à compléter sont les suivantes :

Pour les PC Windows :

SET JAVA_HOME="*C:\Users\Desktop\openlogic-openjdk-jre-8u362-b09-windows-64*"
Renseigner la partie en italique par l'emplacement de votre outil JAVA, que vous avez téléchargé et lors de l'étape 1.

SET REPERTOIRE="*C:\Temp\ENEDIS*"
Renseigner la partie en italique par l'emplacement de vos fichiers zippés et chiffrés, récupérés lors de l'étape 2.

Pour les PC Linux :

export JAVA_HOME=" */usr/lib/jvm/openlogic-openjdk-jre-8u362-b09-windows-64*"
Renseigner la partie en italique par l'emplacement de votre outil JAVA, que vous avez téléchargé et lors de l'étape 1.

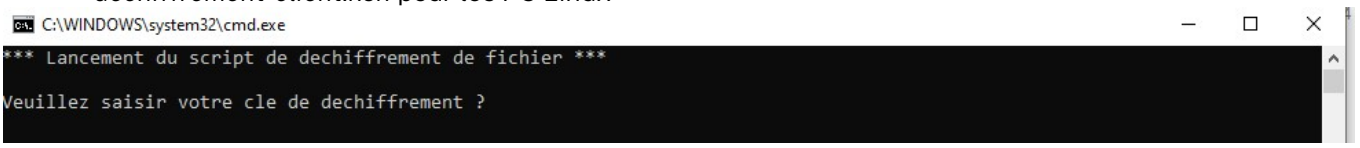
export FILE_FOLDER="*/tmp/ENEDIS*"
Renseigner la partie en italique par l'emplacement de vos fichiers zippé et chiffrés récupéré lors de l'étape 2.

Désormais l'outil est utilisable sur votre poste personnel et vous pourrez passer les étapes précédentes la fois prochaine.

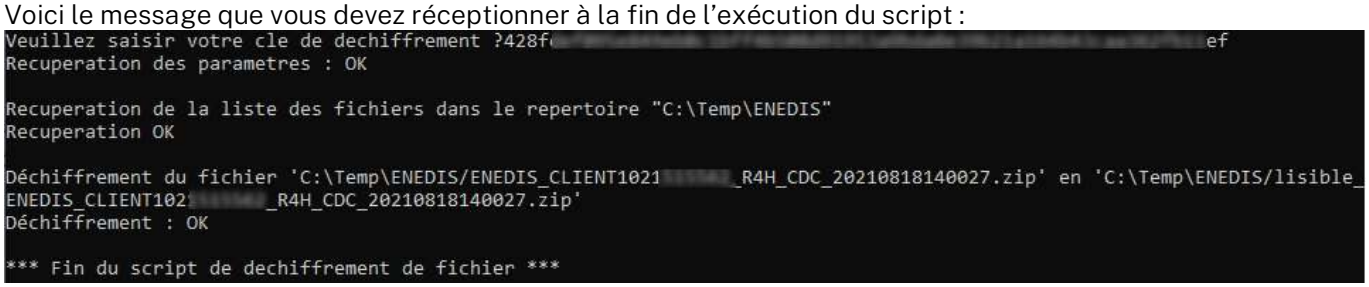
Etape 4 – Utilisation de l'outil :

Il s'agit du lancement de l'outil de déchiffrement : Rendez-vous sur le dossier dézippé « dechiffrement-client ». Puis dans le dossier */bin*. Vous avez alors deux fichiers. Lancez le fichier vous correspondant en double cliquant sur celui-ci

dechiffrement-client.bat pour les PC Windows
dechiffrement-client.ksh pour les PC Linux



Pour utiliser l'outil, entrez la clé de déchiffrement récupérée préalablement conformément au paragraphe 3
Voici le message que vous devez réceptionner à la fin de l'exécution du script :



Vous pourrez consulter vos fichiers désormais lisibles dans le répertoire que vous avez défini.

4.1.4. Exemple de code Java pour le déchiffrement

Si l'outil mentionné précédemment ne convient pas ou ne fonctionne pas, voici un exemple de code JAVA permettant de déchiffrer les fichiers. Cette partie est plus technique que la précédente : JAVA et un IDE sont nécessaires à la bonne compilation du code ci-dessous. Il est aussi nécessaire d'avoir des compétences techniques pour la dérouler.
Le code Java suivant permet de déchiffrer un fichier chiffré en AES256 avec IV (*Initialization Vector*) de 128 bits en en-tête du fichier chiffré.
Il prend 3 paramètres en entrée :

- Le chemin d'accès au fichier chiffré (reçu d'Enedis)
- Le chemin d'accès au fichier déchiffré (qui sera créé)
- La clé symétrique (de chiffrement/déchiffrement) de 256 bits

```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.io.FileOutputStream;
import
java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import
org.apache.commons.codec.binary.Hex;

/**
          * Dechiffre un stream en AES 256 avec la cle keyString
     *
          * @param encryptedStream = Fichier chiffré
          * @param clearStream = Fichier déchiffré
          * @param keyString = Clé AES256 encodée en Hexadécimal (64 caractères)
          * @throws CryptoException
     */

    public static void decryptStream(final InputStream encryptedStream, final
OutputStream clearStream,
                                     final String keyString) throws CryptoException {
try {
             final byte[] keyBytes = Hex.decodeHex(keyString.toCharArray());
             final byte[] key = new byte[keyBytes.length];
             System.arraycopy(keyBytes, 0, key, 0, keyBytes.length);

             final SecretKey keyValue = new SecretKeySpec(key, "AES");

             final Cipher decryptCipher = Cipher.getInstance("AES/CBC/PKCS5Padding",
"SunJCE");
             // Lecture de l'IV depuis le 1er bloc du fichier d'entrée
byte[] iv = new byte[AES256EncryptionUtilDyn.BLOCK_SIZE];
encryptedStream.read(iv, 0, iv.length);
             IvParameterSpec ivspec = new IvParameterSpec(iv);
```

```
            decryptCipher.init(Cipher.DECRYPT_MODE, keyValue, ivspec);

            final byte[] buffer = new byte[1024];
int noBytes = 0;
            final       byte[]      cipherBlock       =     new
byte[decryptCipher.getOutputSize(buffer.length)];
            while ((noBytes = encryptedStream.read(buffer)) != -1) {              final
int cipherBytes = decryptCipher.update(buffer, 0, noBytes, cipherBlock);
                clearStream.write(cipherBlock, 0, cipherBytes);
}

            final int cipherBytes = decryptCipher.doFinal(cipherBlock,
0);              clearStream.write(cipherBlock, 0, cipherBytes);
        } catch (final Exception e) {
            throw new CryptoException("Erreur lors du déchiffrement des données", e);
        } finally {
try {
                if (encryptedStream != null) {
encryptedStream.close();
                }
                if (clearStream != null) {
clearStream.flush();
clearStream.close();
                }
            } catch (final IOException localIOException) {
}
        }
    }
```

4.2. Méthode 7-Zip / Winzip

4.2.1. Méthode de chiffrement

Cette méthode de chiffrement consiste à utiliser la fonctionnalité de chiffrement en mode AES256 offerte par le logiciel libre « 7-Zip » pour chiffrer un document à transmettre par mail au client entreprise (à un interlocuteur désigné par lui), sous la forme d'une « archive » au sens « 7-Zip », qui se présente sous la forme d'un fichier d'extension « .zip ».
La clé de chiffrement est choisie au moment de l'envoi et est communiquée par SMS à l'interlocuteur du client entreprise, qui l'utilisera pour déchiffrer le document (la clé est en effet « symétrique »).
Ainsi, le document chiffré et la clé n'empruntent pas les mêmes canaux de communication.

4.2.2. Méthode de déchiffrement

Lorsque l'interlocuteur de l'entreprise procède à l'ouverture de l'archive, son logiciel de compression (7-Zip ou autre comme par exemple « WinZip ») reconnaît qu'il s'agit d'une archive chiffrée et demande le « mot de passe » : il faut alors saisir la clé reçue par SMS pour obtenir le document « en clair ».