Når klienten sender passwords til serveren skal disse være enkrypteret både via XOR når de sendes mellem klient og server, men også AES 128 enkrypteres når de ligger i databasen, så de ikke ligger i cleartext

Jeg har oprettet et dummy account i createDBscript.sql der hedder:

[admin@admin.dk](mailto:admin@admin.dk)

1234

Koden er allerede hashed til AES 128 i databasen, med nøglen: DIST@DOEK4EVAH!!

Nedenstående kode bruges til hhv. At enkryptere og dekryptere AES 128 med nøglen allerede indsat, i kan bryge denne classe til at enkryptere jeres passwords på klienten inden de sendes til serveren.

**import** java.security.Key;

**import** javax.crypto.spec.SecretKeySpec;

**import** javax.crypto.Cipher;

**import** sun.misc.\*;

**public** **class** **encryptionAES** {

**private** **static** **String** *algorithm* = "AES";

**private** **static** **byte**[] *keyValue*=**new** **byte**[]

{ 'D', 'I', 'S', 'T', '@', 'D', 'O', 'E', 'K', '4', 'E', 'V', 'A', 'H', '!', '!' };

// Performs Encryption

***@SuppressWarnings***("restriction")

**public** **static** **String** **encrypt**(**String** plainText) **throws** Exception

{

Key key = *generateKey*();

**Cipher** chiper = **Cipher**.*getInstance*(*algorithm*);

chiper.init(**Cipher**.ENCRYPT\_MODE, key);

**byte**[] encVal = chiper.doFinal(plainText.getBytes());

**String** encryptedValue = **new** BASE64Encoder().encode(encVal);

**return** encryptedValue;

}

// Performs decryption

***@SuppressWarnings***("restriction")

**public** **static** **String** **decrypt**(**String** encryptedText) **throws** Exception

{

// generate key

Key key = *generateKey*();

**Cipher** chiper = **Cipher**.*getInstance*(*algorithm*);

chiper.init(**Cipher**.DECRYPT\_MODE, key);

**byte**[] decordedValue = **new** BASE64Decoder().decodeBuffer(encryptedText);

**byte**[] decValue = chiper.doFinal(decordedValue);

**String** decryptedValue = **new** String(decValue);

**return** decryptedValue;

}

//generateKey() is used to generate a secret key for AES algorithm

**private** **static** Key **generateKey**() **throws** Exception

{

Key key = **new** SecretKeySpec(*keyValue*, *algorithm*);

**return** key;

}

// performs encryption & decryption

**public** **static** **void** **main**(**String**[] args) **throws** Exception

{

**String** plainText = "1234";

**String** encryptedText = *encrypt*(plainText);

**String** decryptedText = *decrypt*(encryptedText);

**System**.out.println("Plain Text : " + plainText);

**System**.out.println("Encrypted Text : " + encryptedText);

**System**.out.println("Decrypted Text : " + decryptedText);

}

}