OTPme - Feature #45

add backend encryption for all sensitive data

05 April 2015 18:52 - The 2nd

Status:	In Bearbeitung	**Start date:**	19 October 2015
Priority:	Normal	**Due date:**	
Assignee:	The 2nd	**% Done:**	80%
Category:		**Estimated time:**	0:00 hour
Target version:	OTPme 0.3		

Description

we should not save password (hashes), PINs etc. in plaintext.

Subtasks:

Feature # 59: Add header to encrypted attributes **In Bearbeitung**

History

#1 - 05 April 2015 18:53 - The 2nd

- Subject changed from Add backend encryption for all sensitive data to add backend encryption for all sensitive data

#2 - 04 July 2015 02:30 - The 2nd

- Status changed from Neu to In Bearbeitung

- % Done changed from 0 to 90

current implementation uses AES encryption in CFB mode.

```
from Crypto.Cipher import AES
from Crypto import Random

def encrypt(aeskey, data):
""" encrypt string with given aes key """
iv = Random.new().read(AES.block_size)
cipher = AES.new(aeskey.decode("hex"), AES.MODE_CFB, iv)
encrypted_data = iv + cipher.encrypt(data)
return encrypted_data.encode("hex")
```

still needs some investigation if this is the way to go. but replacing the encrypt/decryption functions should be easy.

#3 - 19 October 2015 20:25 - The 2nd

Maybe we should use AES in GCM mode in the future:

- https://bugs.launchpad.net/pycrypto/+bug/899817
- https://github.com/dlitz/pycrypto/pull/33