After decompiling the app with apktool, I started looking through the files. What struck me as odd was the single asset file named private.mp3. Trying to open it resulted in an error, so I decided to investigate further, since it seems important.

```
┌──(kali㉿kali)-[/tmp/VMwareDnD/zUDu66]
└─$ file private.mp3
private.mp3: gzip compressed data, was "private.mp3", last modified: Wed Nov  4
14:28:05 2020, max compression, original size modulo 2^32 22978560
```

Oh, now it all makes sense! The file is actually a .tar.gz so, after extracting, we find among other things a main.py file.

After deobfuscating the code and analizing the logic, we can manage to uncover the flag. The program effectively contains 2 hardcoded encrypted strings: **i**, the password and **g**, the string. The program tries to decrypt the password using the input and checks if the result is the same as the decryption done with the xor key **viafrancetes**. If yes, it decrypts the **g** string using the decrypted key.

```
def n(byt):
    q = b'viafrancetes'
    f = len(q)
    return bytes(c ^ q[i % f] for i, c in enumerate(byt))
def d(s):
    y = n(s.encode())
    return y.decode("utf-8")
encrypted_password =
'R[\x18BCSJ\x10)D+2\x01]6>(\x05\x16R<:T%\x04(X\x0e\x07O\x1aS\x065\x0f"?1\x07$\x02
'
encrypted_g_string =
"\x15\x1d\x07\x1dATX\x00P\x11RJG\r\x04VJW_S\x07L\x00J\x15\x0bQV\x13WZ\x07TB\x06A\
x15\x0f\x02T\x10\x04^S\x07EV@\x10\r\x07\x07GPW[QFUAG]XVK\x02\rR\x18"
decrypted_password = d(encrypted_password)
print(f"Decrypted Password: {decrypted_password}")
decrypted_g_string = d(encrypted_g_string)
print(f"Decrypted 'g' string: {decrypted_g_string}")
```

sAfter running this script we get the flag.

Made with love by: AndreiCat