**Client Authentication**

The Content Generator (contentgenerator project) is a client that sends requests to the Captionify.com server. In this scenario, this is not a “real user”, so the client sessions and cookies aren’t really working as expected.

The server authenticates all requests, such as Post Challenge, by checking if the req.user object exists, which is supposed to indicate that the user is already logged in. In case of Content Generator client, this isn’t the case.

So, to handle this:

* The Content Generator sends the user info for the user that is supposed to be the logged in user in the form of:
  + The user object
  + Digital Signature of the user object (signed using the content generator’s private key)
* When Captionify server receives a request and doesn’t find the req.user object, it looks to see if the private user object along with the digital signature is present
  + If the user object exists, it uses the content generator’s public key to verify the digital signature against the user object, and checks to make sure the two signatures match. If they do, it means it is the content generator who is sending the request

**Generating the Public and Private Keys**

The public and private keys are generated using the openssl commands below.

* Private Key:
  + openssl genrsa -des3 -out ~/private.pem 2048 #generate the private key (this will have the passphrase)
  + openssl rsa -in ~/private.pem -out ~/private\_nopassphrase.pem #remove the passphrase from the private key
* Public Key:
  + openssl rsa -in ~/private\_nopassphrase.pem -outform PEM -pubout -out ~/public.pem #generate the public key from the private key

**Storing the Public and Private Keys**

The two keys are stored in the Credentials > Captionify.com folder as content\_generator\_public\_key.pem and content\_generator\_private\_key.pem

When using the key on the client/server:

* The private key is stored in the client. In Content Generator, the .env file contains the path to the stored private key.
* The public key is stored on the server. In Captionify, the .env file contains the path to the stored public key

**References:**

* http://imadeit-davidjanes.tumblr.com/post/144906098561/nodejs-crypto-signing-and-verifying-messages
* https://rietta.com/blog/2012/01/27/openssl-generating-rsa-key-from-command/
* https://alessiodini.wordpress.com/2009/03/03/generating-certificate-without-passphrase/ (to remove passphrase)