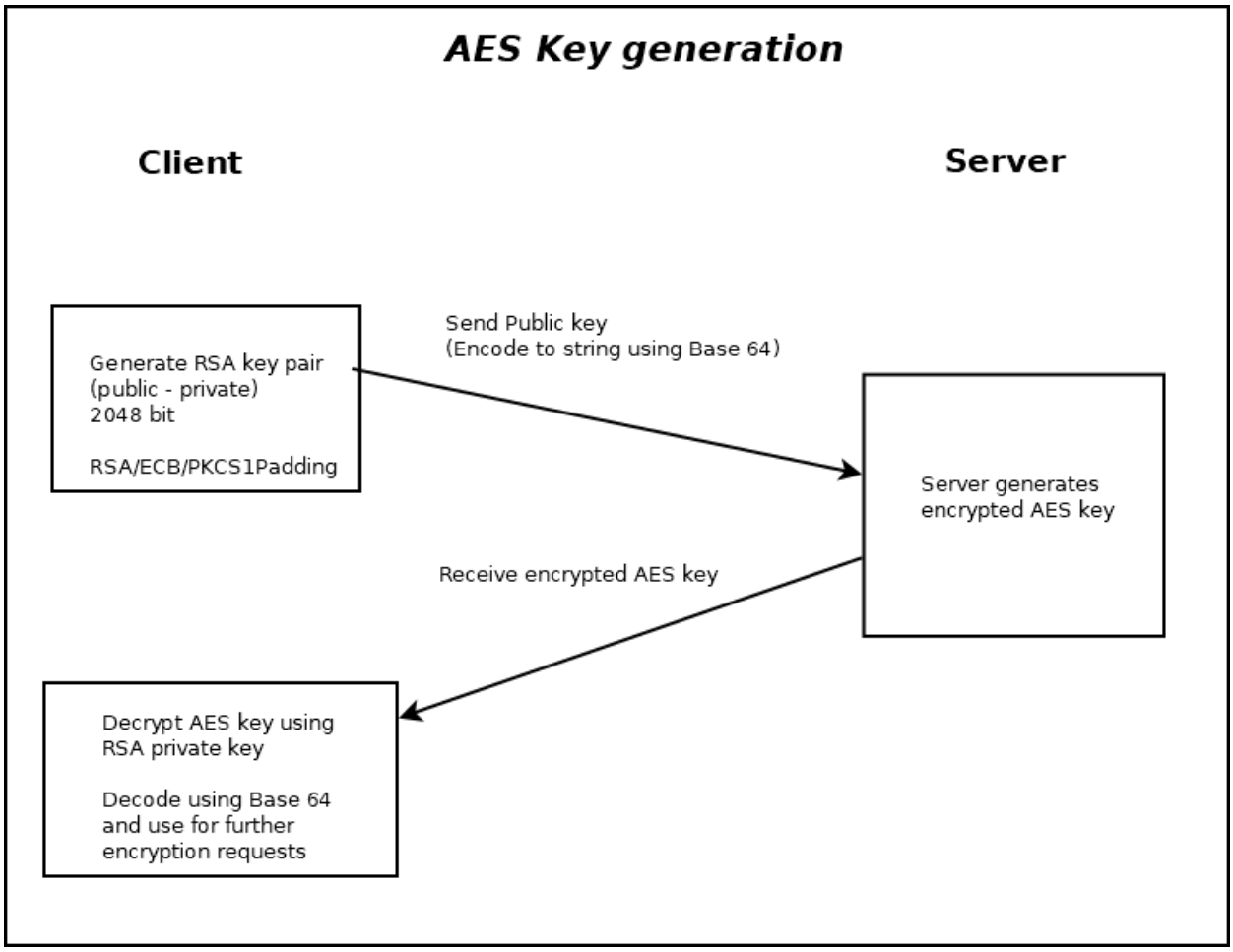
AES Encryption logic

Following is the architecture for fetching AES key during login :



As shown, the mentioned flow is used for fetching the AES key that is to be used for encrypting the request data in all the following requests.

- Algorithm used for RSA is **RSA/ECB/PKCS1Padding**
- Algorithm used for AES is **AES/CBC/PKCS5Padding**
- In the following requests, a random string (IV) is to be used for encryption and decryption with AES. Code to generate IV is provided in Code samples section. IV

Password hashing

- Password is to be hashed using SHA 512 logic. The string is to be hashed 1000 times before sending it in the request.
- A random string (salt) is to be used in the hashing logic. Code has been provided in the Code samples section.

Code Samples

- Create IV :

```
SecureRandom random = new SecureRandom();
   byte iv[] = new byte[16];// generate random 16 byte IV AES is always
                                                  // 16bytes
   random.nextBytes(iv);
```

- Encrypt using AES & IV :

```
Cipher aesCipher = Cipher.getInstance(aesAlgoName);
IvParameterSpec ivspec = new IvParameterSpec(byteIV);
aesCipher.init(Cipher.ENCRYPT_MODE, aesKey, ivspec);
byte[] bytePlainText = aesCipher.doFinal(text.getBytes("UTF-8"));

Encode the byte array to Base 64 string
return Encodedstring
```

- Decrypt using AES & IV :

```
//decode from base 64 string to byte array
byte[] encryptedText = utilObj.getStrFromBase64(text);

Cipher aesCipher = Cipher.getInstance(aesAlgoName);

IvParameterSpec ivspec = new IvParameterSpec(byteIV);
aesCipher.init(Cipher.DECRYPT_MODE, aesKey, ivspec);
byte[] bytePlainText = aesCipher.doFinal(encryptedText);

return new String(bytePlainText);
```

- Password Hash

```
MessageDigest mda = MessageDigest.getInstance(msgDigestName);

byte[] bytArr = password.getBytes("UTF-8");
byte[] salArr = salt.getBytes("UTF-8");

for (int i = 0; i < 1000; i++)
{
   //encode from bytes to Base 64 string
   String str = utilObj.getBase64FromStr(bytArr) +

                //encode from bytes to Base 64 string
                utilObj.getBase64FromStr(salArr);

   bytArr = str.getBytes("UTF-8");

   bytArr = mda.digest(bytArr);
 }
        //encode from bytes to Base 64 string
 return utilObj.getBase64FromStr(bytArr);
```