a) public_variable without prefix seller or buyer means that it’s a current key on client js/mobile app

1. Post profile to the blockchain
cus: encryptContent(stringToHex(data.contentData), data.prvkey1)

2. make offer
buyer_access_string: public_key

3. accept offer
shared_key = getSharedKey(buyer_public_key, private_key)

```
let encryptPass;

if (data.seller_access_string) {  // if user is second, third ... owner
let decryption_shared_key = get_shared_key(data.seller_pubkey, data.prvkey1)
let encryption_shared_key = get_shared_key(data.buyer_pbkey, data.prvkey1)
let pass = decryptPassword(data.seller_access_string, decryption_shared_key);
encryptPass = encryptAccessPassword(pass, encryption_shared_key);
} else {                              // if user is the first owner
let shared_key = get_shared_key(data.buyer_pbkey, data.prvkey1)
encryptPass = encryptPassword(data.prvkey1, shared_key)
}

let message = {
"timestamp": moment().format('YYYYMMDDHHmm'),
cid: data.cid,
coinid: data.coinid,
buyer_access_string: encryptPass,
seller_access_string: encryptPass,
buyer_pubkey: data.buyer_pbkey,
access_type: data.access_type
};
```

4. view user contents

```
getDecryptedContent({ commit }, data) {
if (!data.seller_access_string || data.seller_access_string === 'none') { // if user is the first owner
return hexToString(decryptContent(data.encryptedContent, data.privateKey));
} else {
let shared_key = get_shared_key(data.seller_pubkey, data.privateKey);
let pass = decryptPassword(data.seller_access_string, shared_key);
return hexToString(decryptContentByHash(data.encryptedContent, pass));
}
```