1. A screen shot of a computer

   Description automatically generated
2. Asymmetric key can also be used to sign data.
   1. But in this case, data flow is opposite.

In encryption, the flow was from the outside world to the owner of the Private Key (means public key is used to encrypt data and private key is used to decrypt Data)  
But here in case of signing data, the flow is from the owner of the Private Key to the outside world (means private key is used to sign data and public key is then used to verify the signature of the data).

Jatin: To understand the following point just move to the tag “Jatin001” after reading all the points. So skip it and later on search for “Jatin001”.

1. By signing process, we want a receiver to be 100% sure that this data was received from the owner of the private key whose public key is with the receiver.

A close up of black text

Description automatically generatedBecause even if data is same but when it will be signed using a different private key (as it is unique) then the encrypted hash will be different and when it will be decrypted using some other public key, it will not be able.

1. Signing and Verifying is Two-step process: Remember here we are not focusing on encrypting data but how singing and verifying signature happens.
   1. **From Owner Side**:
      1. The owner creates hash of the data using any hash-function (MD5, SHA).
      2. Then the hash is encrypted using private key -> Output -> Encrypted Hash.
      3. Then Data + Encrypted Hash are sent to the receiver.
   2. **From Public Side**: The public should be sure that this data was sent from the owner of the Private Key. So, it follows the steps:
      1. It takes data to create hash using the same hash-function.
      2. Then it will decrypt the received encrypted hash (called Digital Signature) using the public key.
      3. Then it will compare this newly generated hash with the received one from the owner.
      4. If both matches, it means that data was sent by the owner of the private key.  
         This is because the generated encrypted value by owner can be decrypted only by its corresponding public key.   
         If we try to decrypt using a different public key, it will not succeed and end up with some error message.
2. A diagram of a sign and key

   Description automatically generated
3. A screenshot of a computer

   Description automatically generated  
   <https://stackoverflow.com/questions/18257185/how-does-a-public-key-verify-a-signature>
4. But What we are achieving by signing the data using Asymmetric Key (Private Key)?
   1. Data was not changed during transfer.
   2. 2nd the data was sent by the owner of the private key.
5. This process is called **Data Signature.**
6. So, the owner signs the data using its private key. Basically, it means it encrypts the hash of data with private key.
7. This process is what happens when Server sends to the client **SSL Certificate**.   
   Each certificate **contains signature of the owner of certificate**, and this signature is verified as we discussed.
8. **In next lecture** :  
   Now it is time to discuss RSA Protocol.  
   This protocol utilizes Asymmetric Keys.  
   This protocol is used in certificate communication.