Digital Signature uses **Public Key Cryptography or Asymmetric Cryptography** technique. As we stated before **Public Key Infrastructure** (PKI) is based on two pair of keys, e.g. Private Key and Public Key. Private Key is not shared, only known to the signer and it is used to electronically sign a document. On the other hand, Public Key is known to the public and used to verify that document. A public key algorithm (e.g. RSA) is used to produce these pair of keys [9].

During the generation of digital signature, using an algorithm a signing software creates a **one-way hash value** or **Message Digest** of the document to be signed. Signer then encrypts this hash value with his Private Key and enact his identity. After that, he/she sends the encrypted hash value along with the document to the receiver. (Sometimes send his public key also)

At the time of verification, receiver decrypts messages using the signer's Public Key to retrieve the hash value. This proves the **authenticity** of Signer because Public Key works only if its corresponding Private Key is involved in the signing, which already approving the identity of the signer. Receiver reproduces **one way hash value** of the received document using the same algorithm that signer have already used then compared it with the previous one (hash value after decryption) [14]. If both hash values are matched then it confirms the **integrity** of the messages, e.g. messages or data have not been changed during the transaction. This working scheme is exposed in figure 2.1.

Most widely accepted hash algorithms are Secure Hash Algorithm 1 (SHA -1) and Message Digest (MD5). SHA-1 can create 160-bit hash where MD5 can do it in 128-bit hash [15]. Possibility of matching hash value of two different document is approximately zero. Although PKI is widely accepted method, it has some drawbacks [13].