declaration or certificates of origin), or intangible (e.g. provision of a service) – which is exchanged between participants in the network. It can involve documents, contracts, cryptocurrencies or any other type of asset.

When a transaction is submitted, various processes take place to guarantee the security of the transaction:

- First, the sender generates a key pair, including a public key and a private key. These keys are mathematically related. The public key is made available to the receiver.
- The sender then hashes* the data to be sent, i.e. converts it into a new digital string of a predefined and fixed length using a mathematical function – a hash. Hashing ensures data integrity and prevents forgery.[1] The resulting hash value is encrypted* using the sender's private key. The encrypted hash forms the digital signature* of the data, i.e. the digital fingerprint of an electronic record. It guarantees that the message was created and sent by the claimed sender and was not altered in transit. The sender cannot deny having sent the message.
- The sender then transmits the digital signature together with the plaintext data to participants in the peer-to-peer network – the receivers.

If the sender does not wish other participants in the network to see the message itself, i.e. the plaintext data contained in the documents submitted, (s)he can choose to encrypt the message.

Step 2:

Once the digital signature has been generated and the message has been hashed and encrypted, they are transmitted to participants in the peer-to-peer network – the receivers, also called nodes* – and added to an unvalidated transaction pool.

Step 3: Validation

The validation process differs depending on the type of DLT and the consensus protocol specific to the blockchain or DLT.

Receivers – in the case of permissioned blockchains, authorized nodes – validate the transaction using the sender's public key to decrypt the transaction. A successful decryption confirms that the transaction originates from the claimed sender. The receiver can then verify the integrity of the data by comparing the decrypted hash value sent by the sender with the hash value that (s)he computed when applying the same hash algorithm on the plain data transmitted by the sender. If both hash values