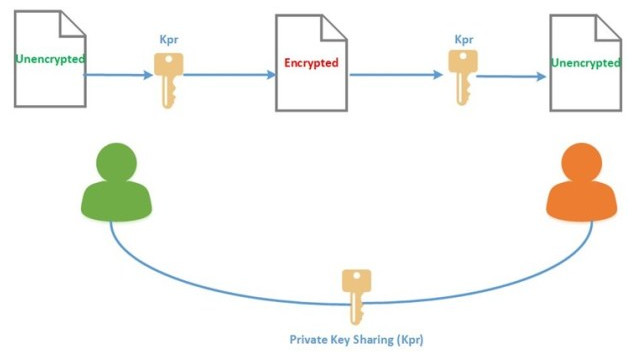
**Cryptography, Encryption, Hash Functions and Digital Signatures**

**Cryptography**

Cryptography is the practice and study of secure communication in the presence of third parties. In the past cryptography referred mostly to encryption. *Encryption*is the process of converting plain text information to cipher text. Reverse is the *decryption.* Encryption is a mechanism to make the information confidential to anyone except the wanted recipients.*Cipher*is the pair of algorithm that creates encryption and decryption. Cipher operation depends on algorithm and the key. *Key* is the secret that known by communicants. In addition, there are two types of encryption by keys used:

**Symmetric Key Algorithms**

Symmetric key algorithms (Private-key cryptography): same key used for encryption and decryption. (AES, DES etc.) (AWS KMS uses Symmetric Key Encryption to perform encryption and decryption of the digital data)

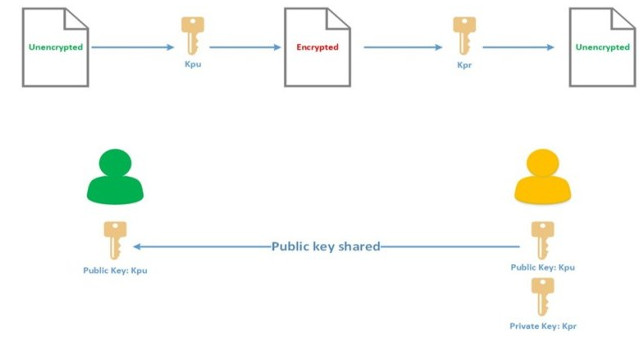


Symmetric key ciphers implemented as either block ciphers or stream ciphers by type of input data. A block cipher enciphers input in blocks of plaintext as opposed to individual characters, the input form used by a stream cipher.

* Block Ciphers: encrypt block of data of fixed size. (DES, AES etc.)
* Stream Ciphers: encrypt continuous streams of data. (RC4, etc.)

Asymmetric Key Algorithms

Asymmetric key algorithms (Public-key cryptography): two different keys (private and public) used for encryption and decryption. (RSA, Elliptic Curve etc.)(AWS EC2 key pairs uses asymmetric encryption.)



Data manipulation in symmetric systems is faster than asymmetric systems as they generally use shorter key lengths. Asymmetric systems use a public key to encrypt a message and a private key to decrypt it. Use of asymmetric systems enhances the security of communication however; it consumes CPU resources heavily.

**Cryptographic Hash Functions**

Cryptographic hash functions are a third type of cryptographic algorithm. A message of any length taken as input, and output to a short, fixed length hash. (MD5, SHA etc.) It is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size (a hash) and designed to be a one-way function, that is infeasible to invert. Integrity checking is the mechanism to verify if the information has not changed. To validate the integrity, a thumbprint (also called hash or digest) of the information created. Thumbprint created by an algorithm that create a shorter bit string from an information.



**Digital Signature**

Digital signature is a mathematical scheme for demonstrating the *authenticity* of digital messages or documents. A valid digital signature enables information integrity (using hash algorithm) to ensure message is not altered, message created by the sender (authentication) and sender cannot deny having sent the message (non-repudiation). The digital signature has to be authentic, unfalsifiable, non-reusable, unalterable and irrevocable. When all this properties are gathered, the authenticity and the integrity of an information can verified.

**how digital signatures are used in blockchain**

Digital signatures play a critical role in ensuring the authenticity and integrity of transactions on a blockchain. Here's how digital signatures are used in blockchain:

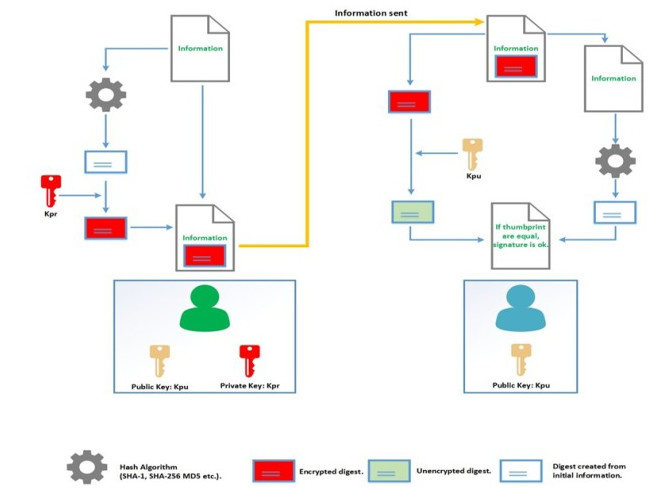
Signing transactions: When a user wants to send a transaction on a blockchain, they use their private key to sign the transaction. This creates a unique digital signature that can only be created with the user's private key.

Verification: To verify the authenticity of the transaction, other nodes on the network use the sender's public key to verify the digital signature. If the signature is valid, the transaction is considered to be authentic and is added to the blockchain.

Tamper-proofing: Once a transaction is added to a block on the blockchain, it cannot be modified without invalidating the digital signature. This makes the transaction tamper-proof and ensures that it cannot be altered without detection.

Non-repudiation: Digital signatures also provide non-repudiation, which means that the sender cannot deny that they sent the transaction. Since the digital signature can only be created with the sender's private key, it provides proof that the sender authorized the transaction.

Overall, digital signatures are an essential component of blockchain technology, ensuring the security, authenticity, and non-repudiation of transactions on the blockchain.



**Role of Digital Signatures in Blockchain**

**Ensuring Authenticity**

Digital signatures are used to authenticate the identity of the sender of a transaction on the blockchain. When a user wants to make a transaction, they must sign it using their private key. The private key is a unique piece of information used to create the digital signature and is known only to the user who owns it. The digital signature is then attached to the transaction and is used to verify its authenticity when broadcast to the network. By using digital signatures, it is possible to verify that a transaction was made by the person who claims to have made it.

**Providing Non-repudiation**

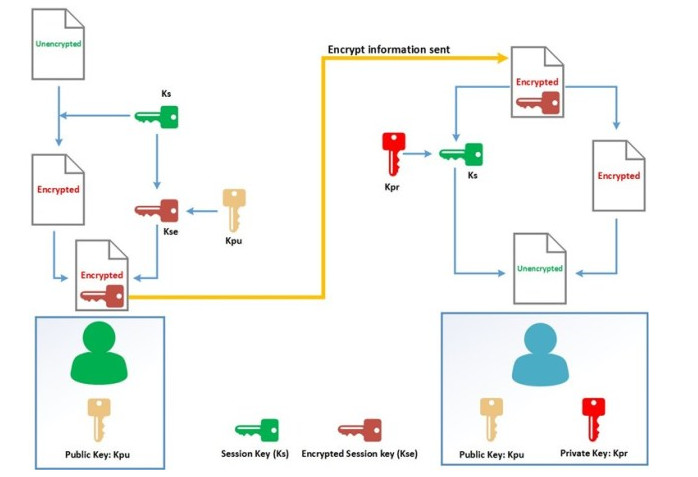
Digital signatures also provide non-repudiation, meaning that the transaction’s sender cannot deny having made the transaction. Once a transaction is signed with a digital signature, it is irreversible, and the sender cannot deny having made the transaction. This helps to ensure the integrity of the data stored on the blockchain, as it is not possible for someone to alter or delete a transaction once it has been made.

**Ensuring Data Integrity**

Digital signatures also help to ensure the integrity of the data stored on the blockchain. When a transaction is signed with a digital signature, it is cryptographically secure, which means altering the transaction without being detected is complicated. This helps ensure that the data stored on the blockchain is accurate and reliable and helps prevent fraud and tampering.

**Modern Encryption**

Each encryption algorithm has advantages and convenient therefore Modern Encryption associates both symmetric and asymmetric techniques. Modern algorithm uses a session key (temporarily key) to encrypt information with symmetric cryptography. Next, the session key encrypted with the public key of the recipient. To unencrypt information, first the recipient unencrypts the session key with his private key and unencrypt information with the session key.



On the sender side, following actions performed:

1.      A temporarily key called session key (Ks) generated;

2.      Information encrypted with session key (Ks);

3.      (Ks) Encrypted with the public key (Kpu) of the recipient. This key called Kse;

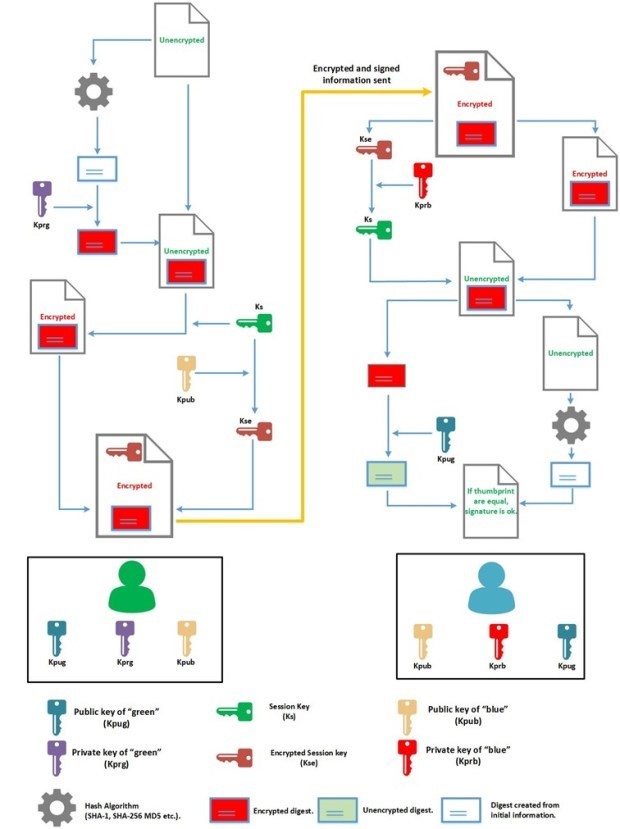
4.      Kse added to the encrypted information file. This file sent to the recipient.

On the recipient side, the below action performed:

1. The encrypted information and Kse are separated;
2. The Kse key is unencrypt with the private key (Kpr) of the recipient and becomes the Ks;
3. The document is unencrypted with Ks.

Encryption and Digital Signature Operation

Now that we are aware about encryption, hash algorithm and signature, let have a look how these elements interact together to make an information confidential, authentic and honest.



When the signature and encryption used together, the signing process done first. Following steps performed:

1. A digest is created from the initial information;

2. This thumbprint is encrypted with the private key (Kprg);

3. The thumbprint is added to the initial information (in the same file);

4. A temporarily session key is generated (Ks) It will be used to encrypt initial information;

5. The session key is encrypted (Kse) with the public key of the rececipient (Kpub);

6. Kse added to encrypted information file. So this file is contains the encrypted information, the Kse and the signature.

When the recipient receives the file from the issuer, it begins by unencrypt file and next to verify the signature:

1. The recipient extract the Kse from the received file. This key is unencrypt with the private key (Kprb) to obtain session key (Ks);

2. Ks is used to unencrypt information;

3. Next recipient extract the encrypted thumbprint;

4. The public key (Kpug) is used to unencrypt the thumbprint;

5. In the same time, the recipient creates a digest from the previously unencrypted information;

6. To finish, the recipient compares the unencrypted thumbprint with the digest generated from unencrypted information. If they match, the signature verified.