Blockchain technology, it is a technology that mixes distributed data capabilities with the protection that unauthorized individuals will not be able to find the data delivered in a single transaction, and because of that it's among the most creative solutions for supporting organizations in protecting the security of the data.

It employs cryptography to achieve the appropriate level of security while also stressing the importance of hashing and digital signatures in blockchain. In the blockchain world, combining digital signature and hashing play a significant role. The following discussion will focus on the significance of digital signatures in the context of blockchain.

As a result, cryptography has emerged as a critical option for protecting the security of sensitive data. Before delivering a message to a recipient, cryptography scrambles the original material into a cipher. The recipient could employ keys to decrypt the cipher, and the keys are only available to him. As a result, no third party could detect the message as it traveled from sender to recipient.

Digital signatures, which are normally used to authenticate the legitimacy of operations, are significantly used in blockchains. Users must show to one another that they can make a transaction, while simultaneously preventing unauthorized individuals from doing so. One of the most signature schemes that is used in blockchain is ECDSA.

ECDSA stands for the Elliptic Curve Digital Signature Algorithm which is currently in use for the as a signature algorithm in use in Bitcoin. In comparison to RSA, ECDSA uses shorter keys and needs fewer process resources whereas maintaining smart security; ECDSA conjointly employs elliptic curves over finite fields. Consider an elliptic curve group as a constrained set of points on a curve where one operation is easy in one direction but difficult in the other.

This algorithm consists of four main requirements:

1- The first element is that the universal domain variables, that specify an elliptic curve and a point of origin on the curve, are used by all users in the digital signature process.

2- The second one is that a sender should first create a pair of keys (public/private) key, then the sender will use this private key to sign the message and the public key will be used by the other side to verify the message.

3- The third one is that for message to be signed a hash value must be generated. This means a signature is created via the private key, the domain parameters, and the hash value.

4- The fourth one is for the message to be verified by the receiver they will use the public key, the domain settings, and the integer x as input. The output is a value y that is compared to w; if w = y, the signature is valid.

The Process of the ECDSA will be as follows:

1- Key Generation:
An ECDSA authenticator requires knowledge of its private key to work. The secret key and domain parameters are used to generate the public key. The pair of keys must be remembered by the identifier. The secret key, as its name implies, is not accessible from the outside world. The public key, on the other hand, must be freely accessible. The production of the key pair is shown in Figure [1]. A random number generator is generated, and after it is finished, the numeric value that forms the secret keys d is given (a scalar). The public key Q (x, y) is then obtained using point multiplication according to Equation: $Q(x, y) = d \times G(x, y)$.

2- Signature Computation:
The signature is represented into two values (s) & (r), so the pair (r, s) together is your ECDSA signature. To calculate the (s) value we need to use a secure hash method, that will represent the input as a message digest H(m). The following are the characteristics of a secure hash: irreversibility, determining the message from its digest will be computationally infeasible, Collision, resistance finding more than one message that can creates a given hash, and large avalanche effect, any change in the message causes a considerable change in the hash. Following the computation of the message digest, a random number generator is used to generate a value k for the elliptic curve computations. As shown in Figure [2].

   ➔ Here we will generate a random number (k), such that (x, y) = k*G, and a random secret number (r), such that r=x mod n.

   ➔ In this step using a secure hash we the variable-length message is first changed to a fixed-length message digest H(m), e=H(m),

   ➔ After r is successfully computed, s is computed according to equation using scalar operations. Inputs are the message digest h(m); the private key d; r; and the random number k: $s = (k^{-1} (h(m) + d * r) \bmod n$, signature.

3- Signature Verification:
The signature verification process is the inverse of the signature computation process. Its goal is to use the senders' public key to validate the message's security. The message digest signed by the authenticator is computed using the same secure hash technique as in the signature step, which, when combined with the public key Q (x, y) and the digital signature components r and s, gives the result. As shown in Figure [3].

   ➔ These equations show the verification process. Inputs are the hash value h(m), the public key Q (x, y), the signature components r and s, and the base point G (x, y):

e=H(m), hash value
$w = s^{-1} \bmod n$
u1 =e*w mod n
u2 =r*w
X= (x2, y2) = u1*G + u2 * Q

ECDSA has an advantage over other public key cryptography in that it is newer. ECDSA was standardized in 2005, whereas RSA, the most widely used public key cryptography method, was standardized in 1995. Because ECDSA has only been around for a short period, cybercriminals have had less time to research how to break it. This, along with the complexity of ECDSA, makes it a more enticing option. Most systems favor ECDSA to RSA for public key cryptography functions because of these benefits. ECDSA has the disadvantage of being difficult to set up, whereas RSA is more straightforward. The ease of use of RSA is frequently a draw for businesses, as it has fewer hurdles in its setup. The poor implementation of ECDSA itself, which is complicated to implement in the first place, has been the downfall of many different businesses that have been hacked.

Appendix:

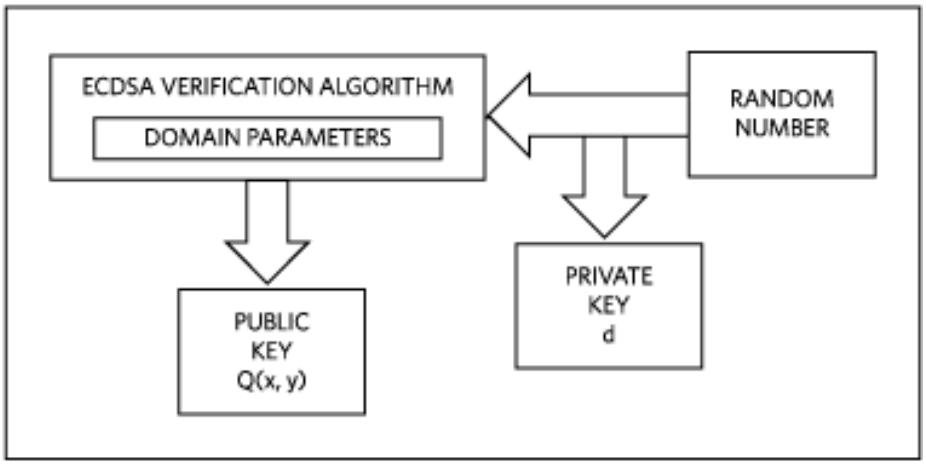

Figure 1:Key pair generation process.

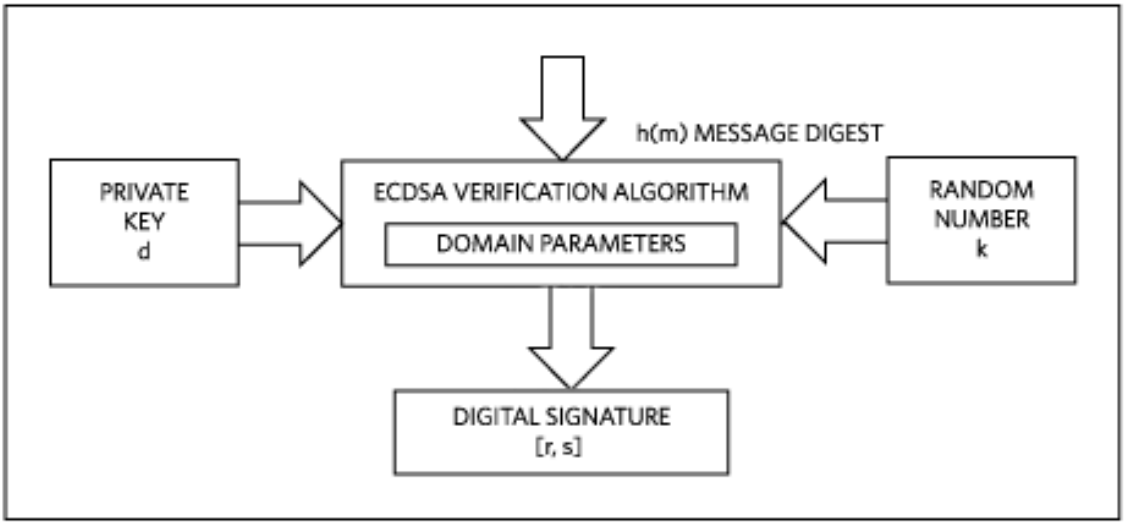

Figure 2:Signature computation process.

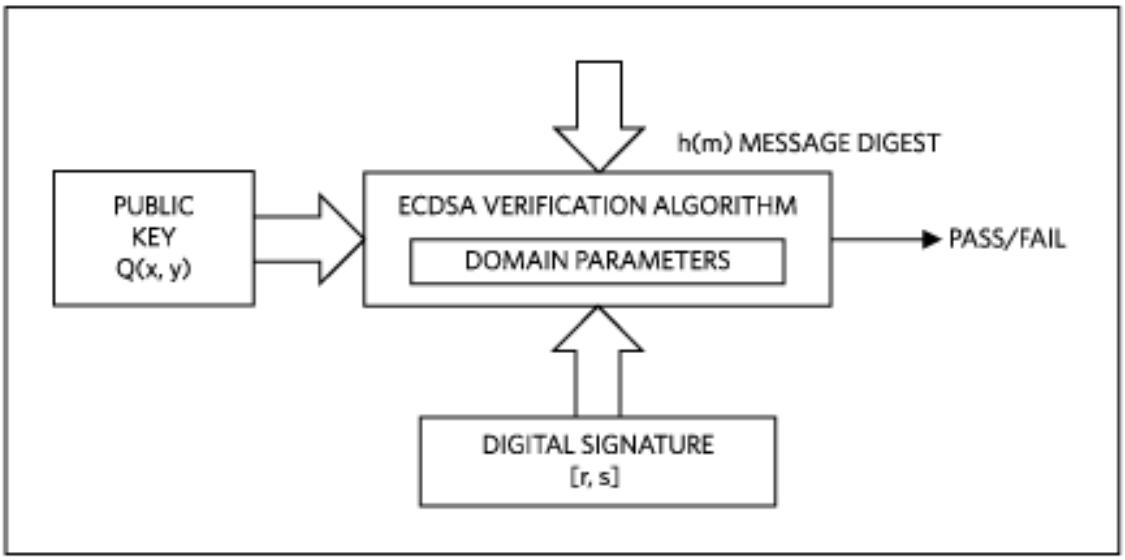

Figure 3:Signature verification process.

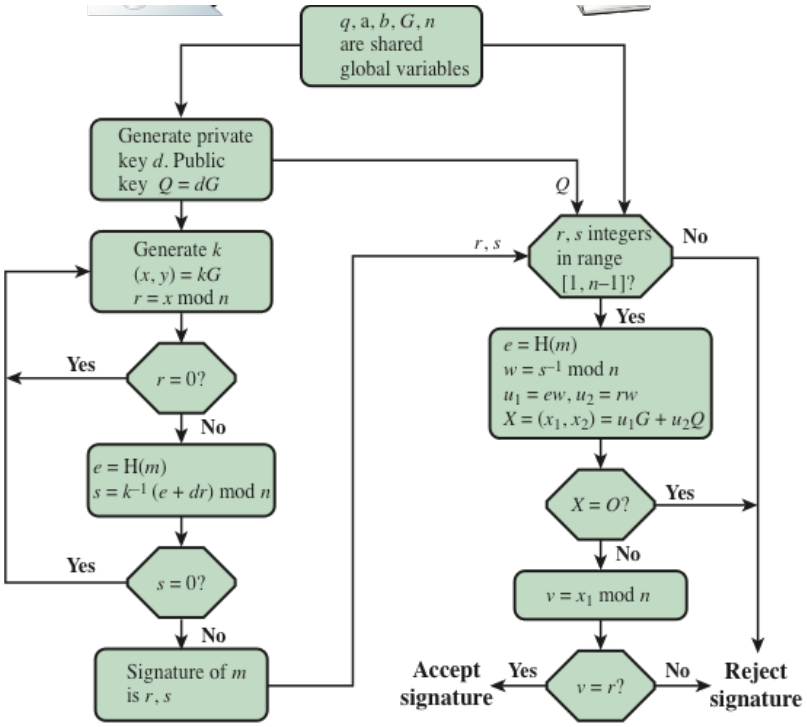

Figure 4:ECDSA Signing and Verification.

Bibliography:

1- https://101blockchains.com/hashing-and-digital-signature-in-blockchain/
2- https://techbriefly.com/2021/05/28/what-are-hashing-and-digital-signature-in-the-blockchain/
3- https://blocchainexpert.medium.com/what-is-hashing-and-digital-signature-in-blockchain-b35bee264002
4- https://coincentral.com/how-digital-signatures-and-hashing-protect-your-transactions/
5- https://bisontrails.co/digital-signatures/
6- https://www.encryptionconsulting.com/education-center/what-is-ecdsa/
7- https://www.maximintegrated.com/en/design/technical-documents/tutorials/5/5767.html