THE FUTURE BANKS

INTRODUCTION

We lack a clear and transparent system when it comes to banking systems. The banks charge a very hefty interest rate on a particular transaction. Banks are very greedy in their profit as they give very less interest rate to the customer and charge very much on their lending.

So to solve this problem of centralized financing system, Mike Ray McDonald and Fernando Martinelli came up with a solution of decentralized financing or "DeFi".

With crypto lending or borrowing we can convert a crypto asset to a term called as "stablecoin"(we will know about this later) which can be pegged to the US Dollar in turn providing financial stability to our coin.

Flash loansFlash loans in DeFi require minimal paperwork and utilize smart contracts for repayment.

DeFi is controlled by the community rather than a central authority.

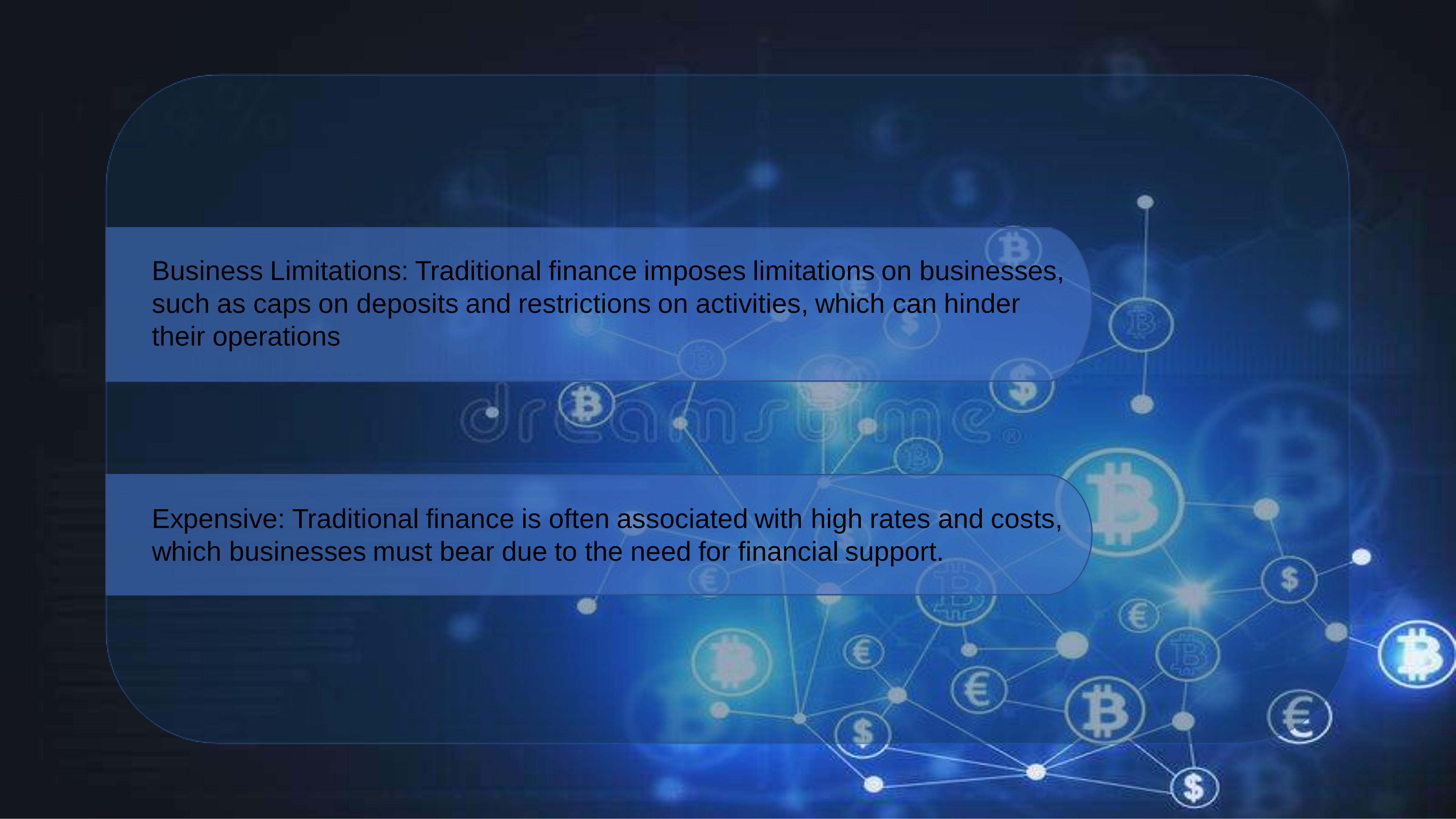
The functioning of decentralized finance will be explored, along with the relation of qubits to blockchain technology.

Safety measures, including algorithms, protect DeFi against attacks, specifically brute force attacks.

DECENTRALIZED FINANCING

Centralized Finance: In centralized finance, there is a central authority (government and banks) that controls the flow of money, can print more money, and has control over borrowing, banking access, and monetary policies.

Trust and Ownership: In traditional finance, proving ownership of money can be challenging as it relies on trust in the central authority. The control over money rests with the centralized entities.

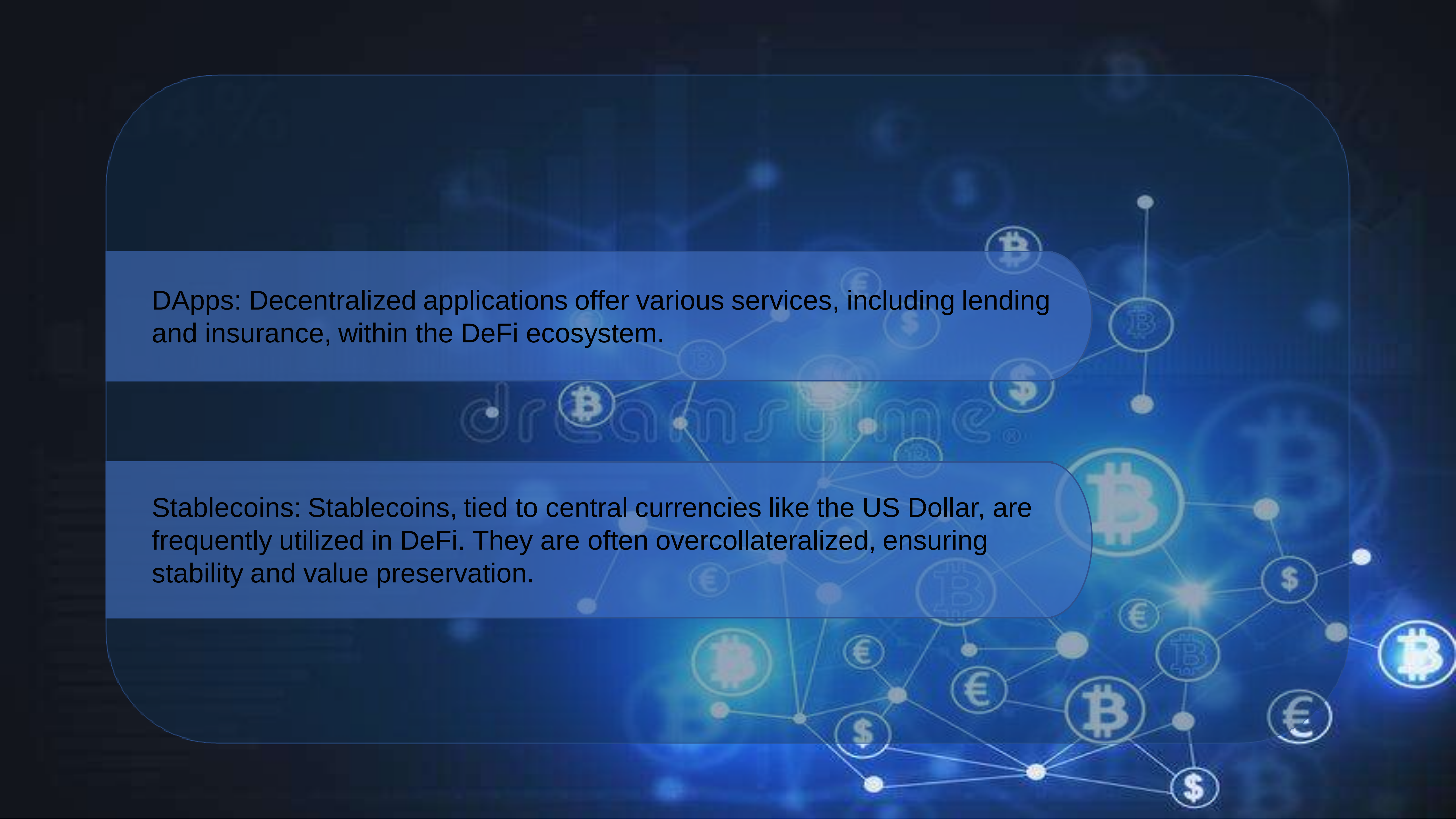


Business Limitations: Traditional finance imposes limitations on businesses, such as caps on deposits and restrictions on activities, which can hinder their operations

Expensive: Traditional finance is often associated with high rates and costs, which businesses must bear due to the need for financial support.

Decentralized Finance (DeFi): DeFi relies on cryptography, blockchain technology, and smart contracts to provide financial services and applications.

Cryptocurrency and Ethereum: DeFi predominantly operates on cryptocurrencies, with Ethereum being a widely used platform. Ethereum enables the creation and execution of decentralized applications (DApps).



DApps: Decentralized applications offer various services, including lending and insurance, within the DeFi ecosystem.

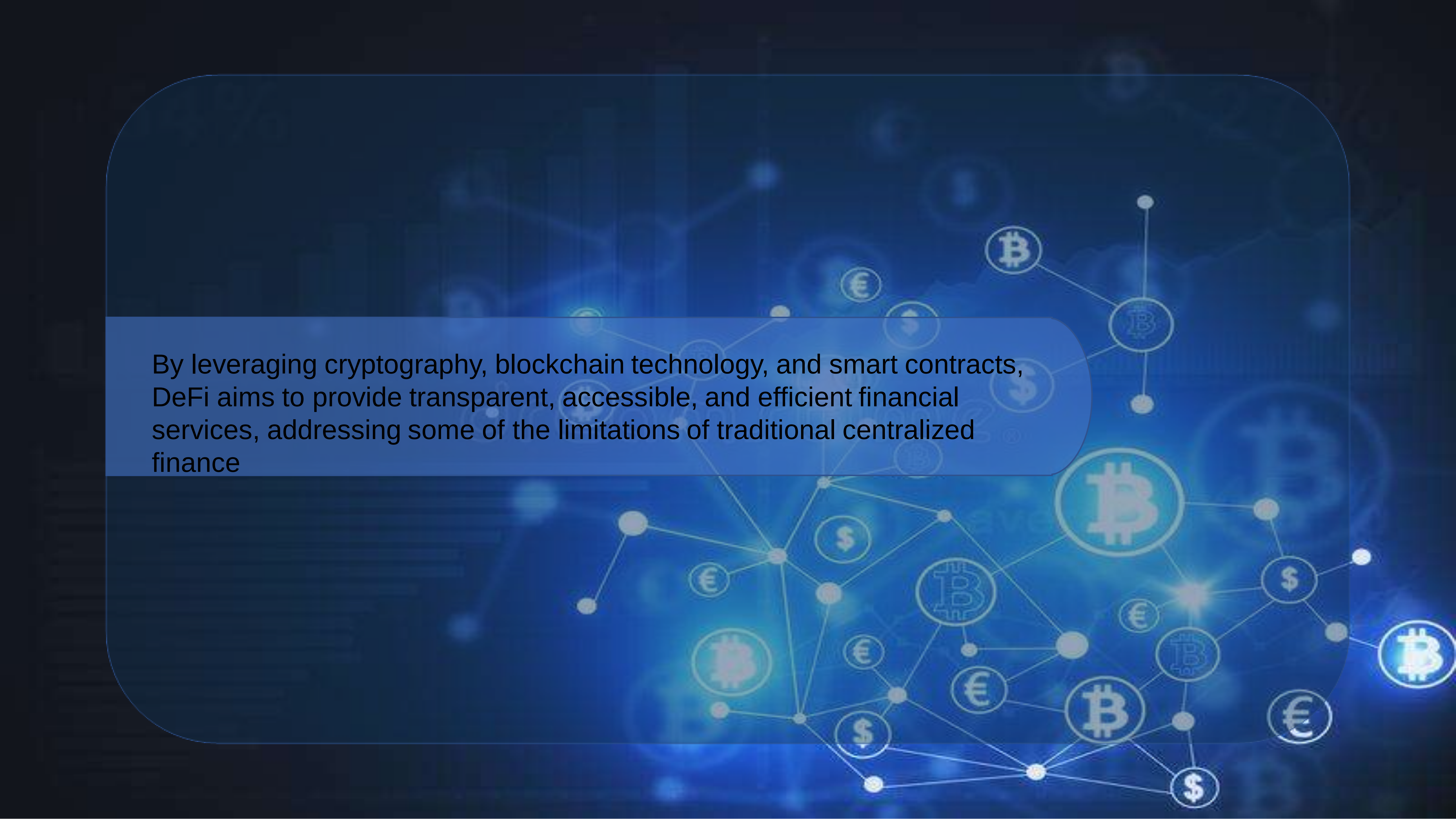
Stablecoins: Stablecoins, tied to central currencies like the US Dollar, are frequently utilized in DeFi. They are often overcollateralized, ensuring stability and value preservation.

Smart Contracts: Smart contracts are self-executing contracts coded with predefined conditions. They operate based on the fulfillment of these conditions, ensuring automated and trustless execution

Blockchain Storage: Smart contracts and other relevant data can be stored in blocks within the blockchain, enabling secure execution and transparency.

Censorship Resistance: DeFi systems are designed to be resistant to censorship, meaning they are less likely to be controlled or manipulated by a central authority.

Cost Efficiency: DeFi generally offers more cost-effective financial solutions compared to traditional centralized finance, potentially reducing fees and expenses for users



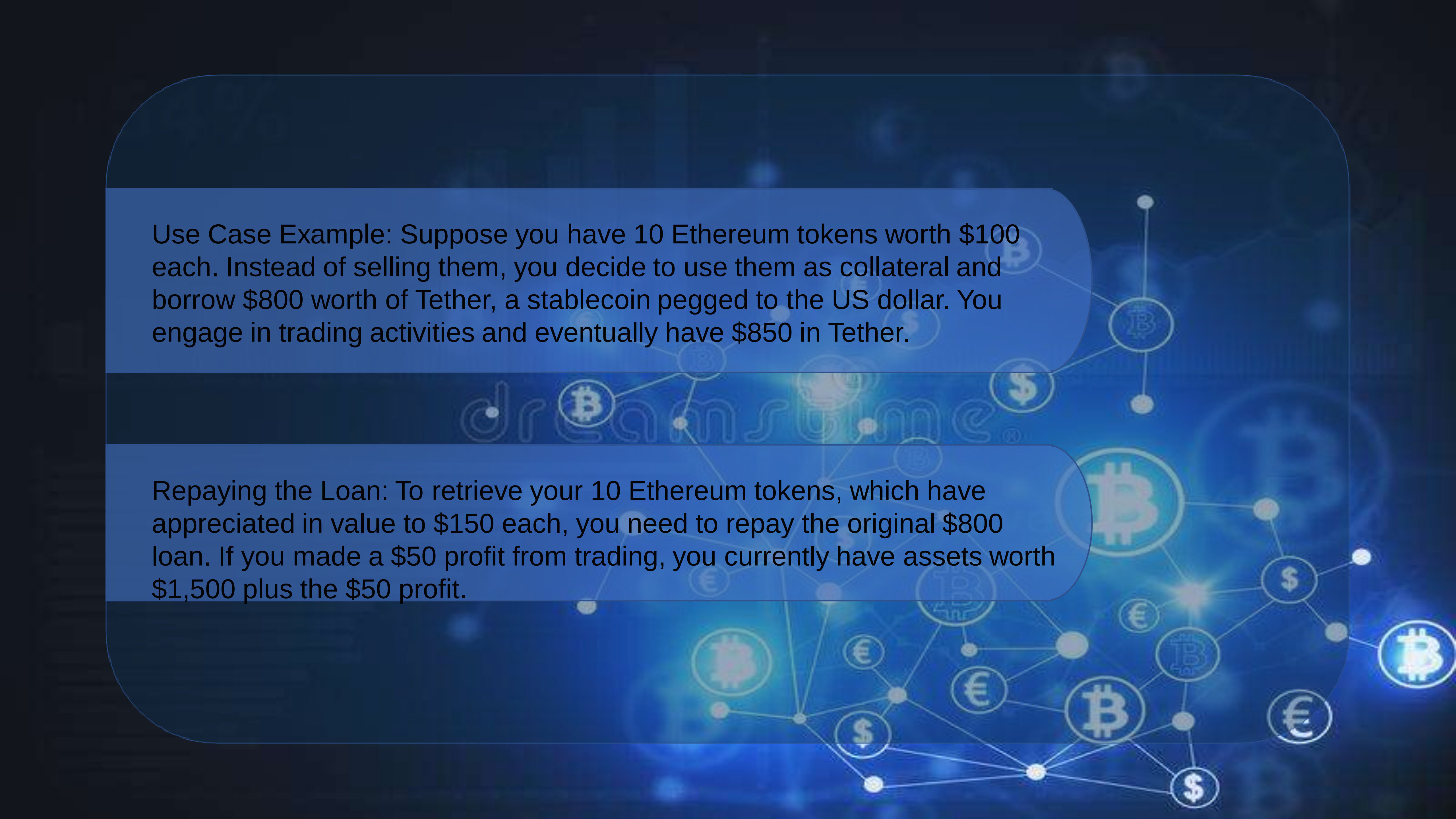
By leveraging cryptography, blockchain technology, and smart contracts, DeFi aims to provide transparent, accessible, and efficient financial services, addressing some of the limitations of traditional centralized finance

SMART CONTRACTS

Need for Custody in Crypto: Unlike traditional loans where legal consequences exist for non-payment, cryptocurrencies lack a mechanism to prevent borrowers from fleeing after borrowing. There is a need for a system to ensure custody of coins during lending and borrowing in the crypto space.

Introduction to Smart Contracts: Smart contracts are pieces of code that execute based on predefined conditions. They function similar to if-else loops in programming languages. These contracts can be stored in blocks and executed at specific times.

Versatile Applications: The algorithm described in the previous section has a variety of applications, including storing smart contracts, codes, or addresses of stablecoins, among others.

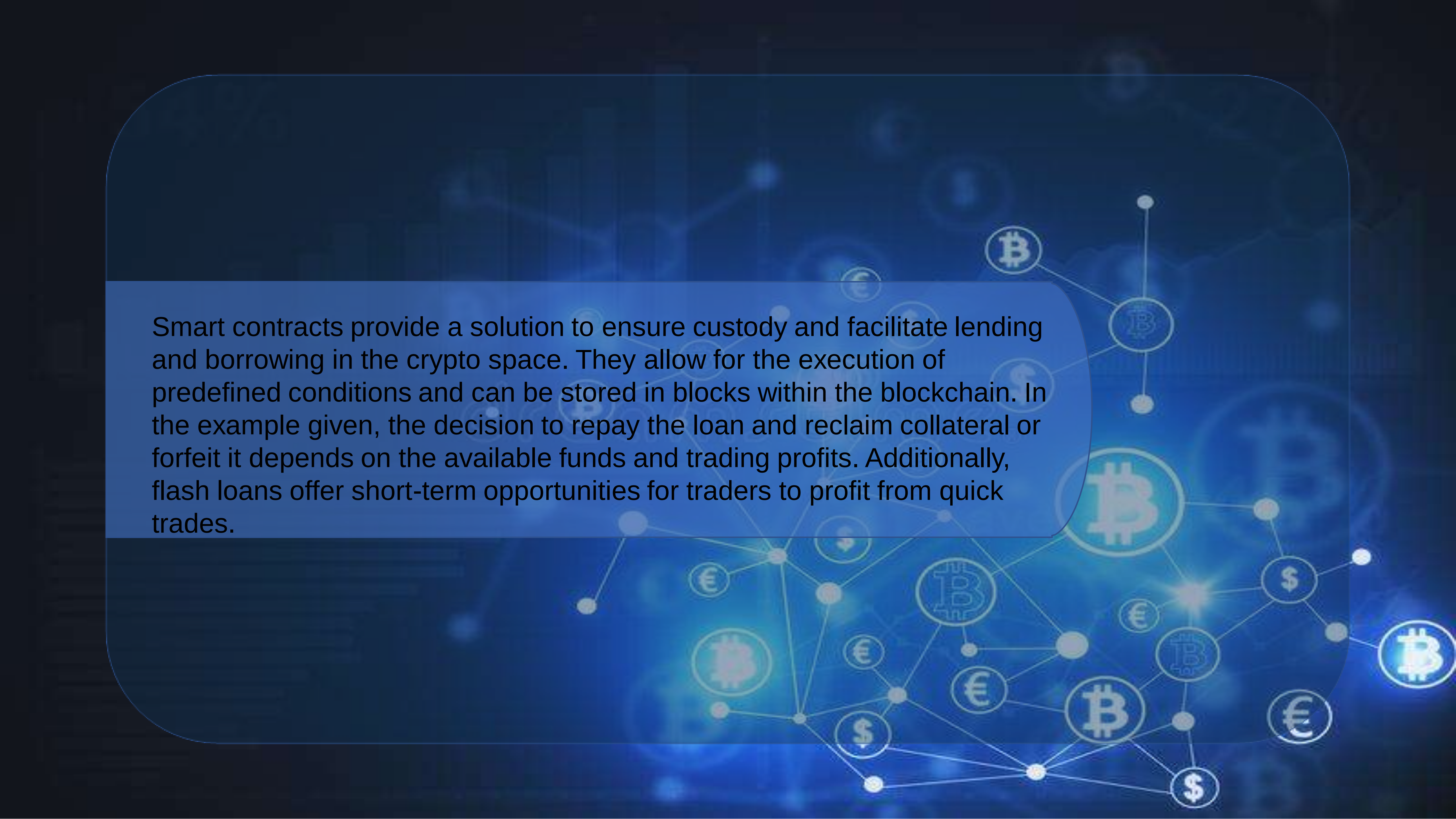


Use Case Example: Suppose you have 10 Ethereum tokens worth $100 each. Instead of selling them, you decide to use them as collateral and borrow $800 worth of Tether, a stablecoin pegged to the US dollar. You engage in trading activities and eventually have $850 in Tether.

Repaying the Loan: To retrieve your 10 Ethereum tokens, which have appreciated in value to $150 each, you need to repay the original $800 loan. If you made a $50 profit from trading, you currently have assets worth $1,500 plus the $50 profit.

Options for Repayment: If you have $850 in Tether, you can add an extra $50 to repay the full loan of $800 and reclaim all your collateral (Option A). Alternatively, if you only have $750 in Tether, you can choose to keep the $750 but forfeit your 10 Ethereum tokens (Option B).

Flash Loans: Flash loans are another type of loan in the crypto space, which have a very short duration, such as 10 seconds. Traders can execute quick trades to profit from price discrepancies between different platforms.

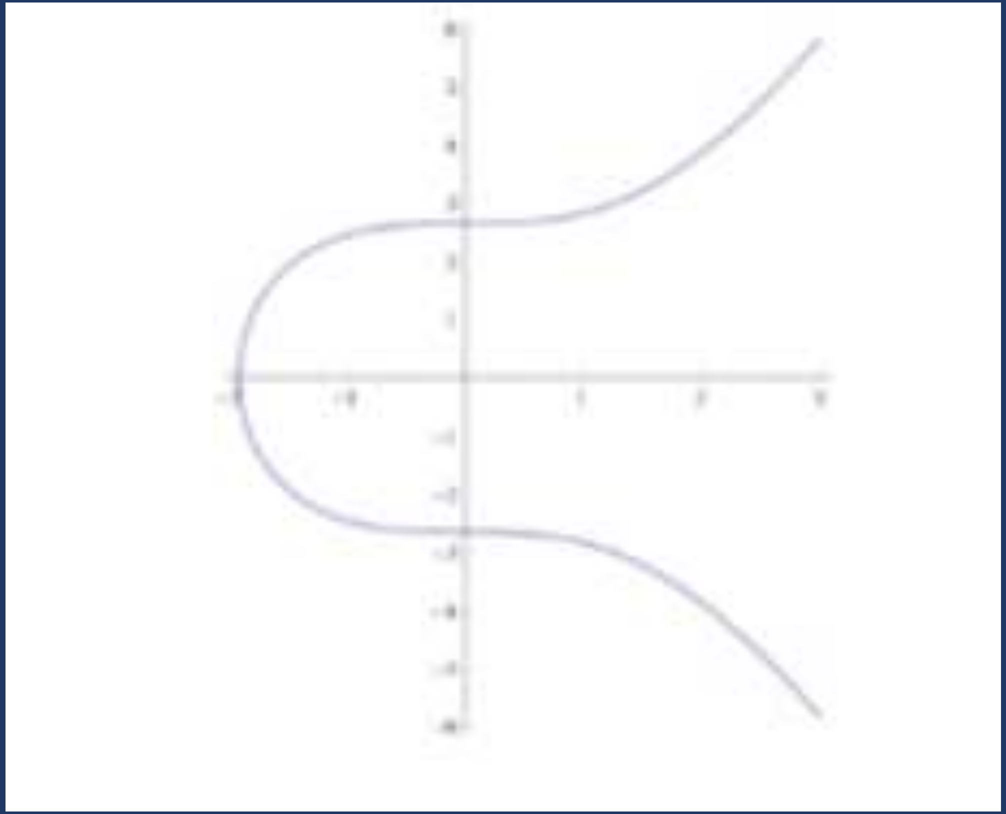


Smart contracts provide a solution to ensure custody and facilitate lending and borrowing in the crypto space. They allow for the execution of predefined conditions and can be stored in blocks within the blockchain. In the example given, the decision to repay the loan and reclaim collateral or forfeit it depends on the available funds and trading profits. Additionally, flash loans offer short-term opportunities for traders to profit from quick trades.

ELLIPTIC CURVE

An elliptic curve is defined by the equation $y^2 = x^3 + ax + b$, where a and b are real coefficients and there are no multiple roots.

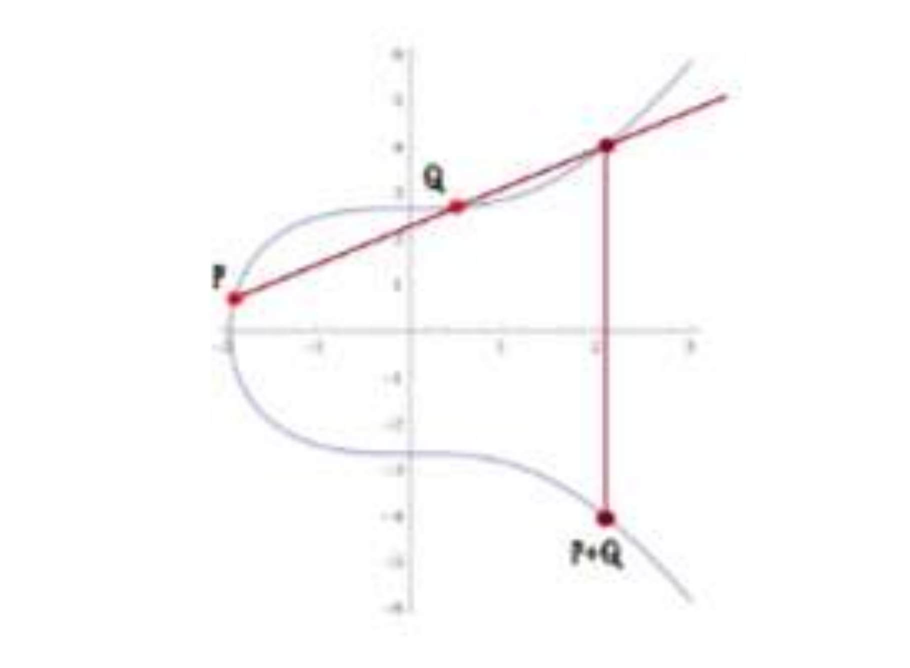
Points on an elliptic curve can be obtained by assigning real values to x and checking if the right side of the equation is a square in R (a real number greater than or equal to 0).



GRAPH OF AN ELLIPTIC CURVE

The point at infinity, denoted as O, is the neutral element of the sum operation on the elliptic curve.

The sum of two different points P and Q on the curve is obtained by drawing a straight line passing through P and Q, finding the intersection point of this line with the curve, and reflecting it symmetrically with respect to the x-axis.



SUM OF DIFFERENT POINTS IN AN ELLIPTIC CURVE

The process described allows for the doubling of a point on the curve.

The set of points on an elliptic curve with the defined sum operation forms an abelian group, satisfying the associative property and having a neutral element (O) and a symmetric element for every point.

Analytical expressions for calculating the sum and double of a point on an elliptic curve can be derived.

For two points P = (x_1, y_1) and Q = (x_2, y_2) on the curve, the sum P + Q = (x_3, y_3) is calculated using specific formulas for x_3 and y_3.

For the double of a point P, 2P = (x_3, y_3), the formulas for x_3 and y_3 are different from those used for the sum operation.

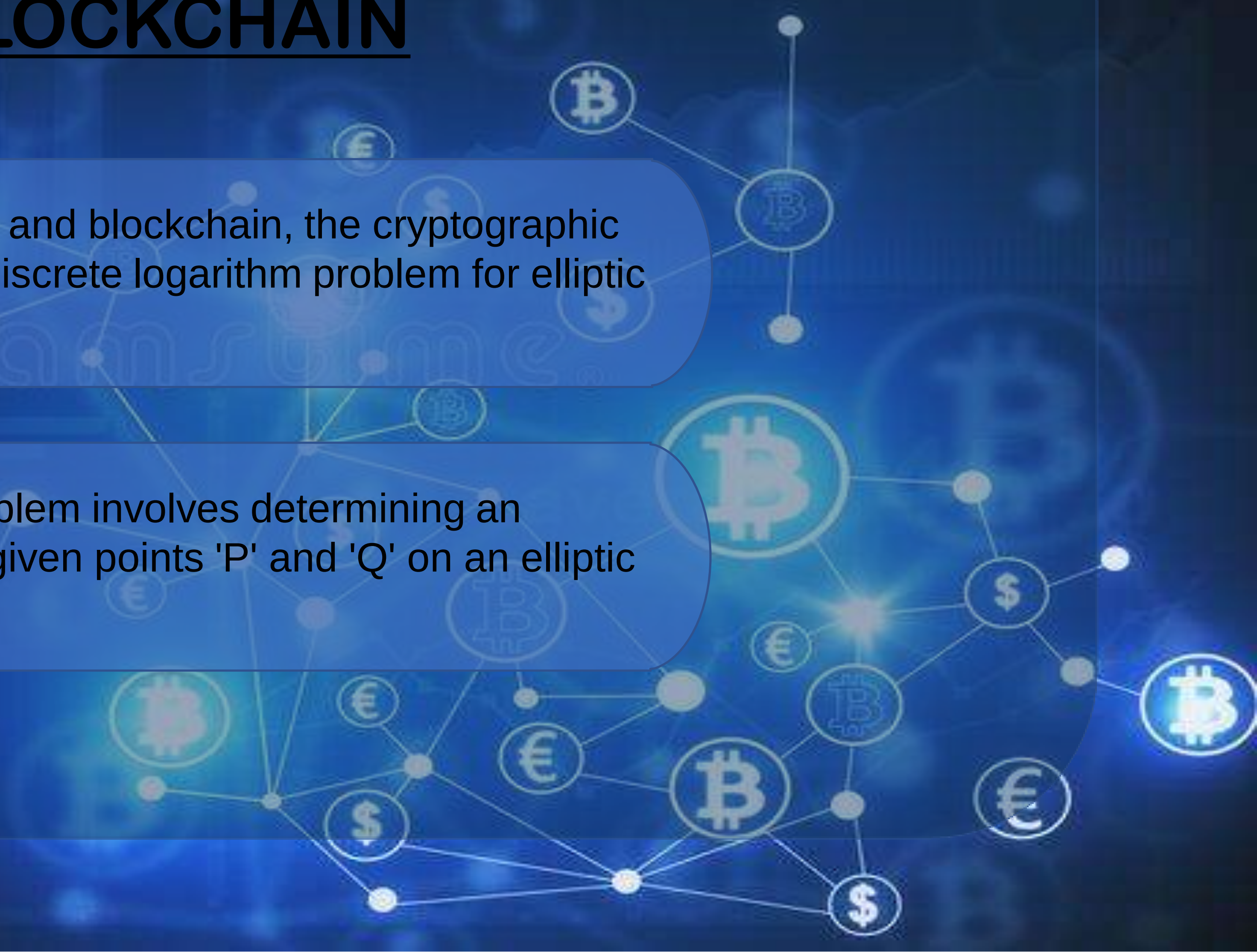
Elliptic curves can be generalized for the case of a finite field Fq, where the characteristic field is other than 2 and 3.

The elliptic curves on a finite field are defined by the equation $y^2 = x^3 + ax + b$, with the polynomial $x^3 + ax + b$ not having multiple roots and including the point at infinity (O).

The sum and product operations on elliptic curves over a finite field are defined using the formulas mentioned earlier.

The set of points on an elliptic curve with the defined sum operation forms an abelian group structure.

CRYPTOGRAPHIC ELEMENTS IN BITCOIN AND BLOCKCHAIN

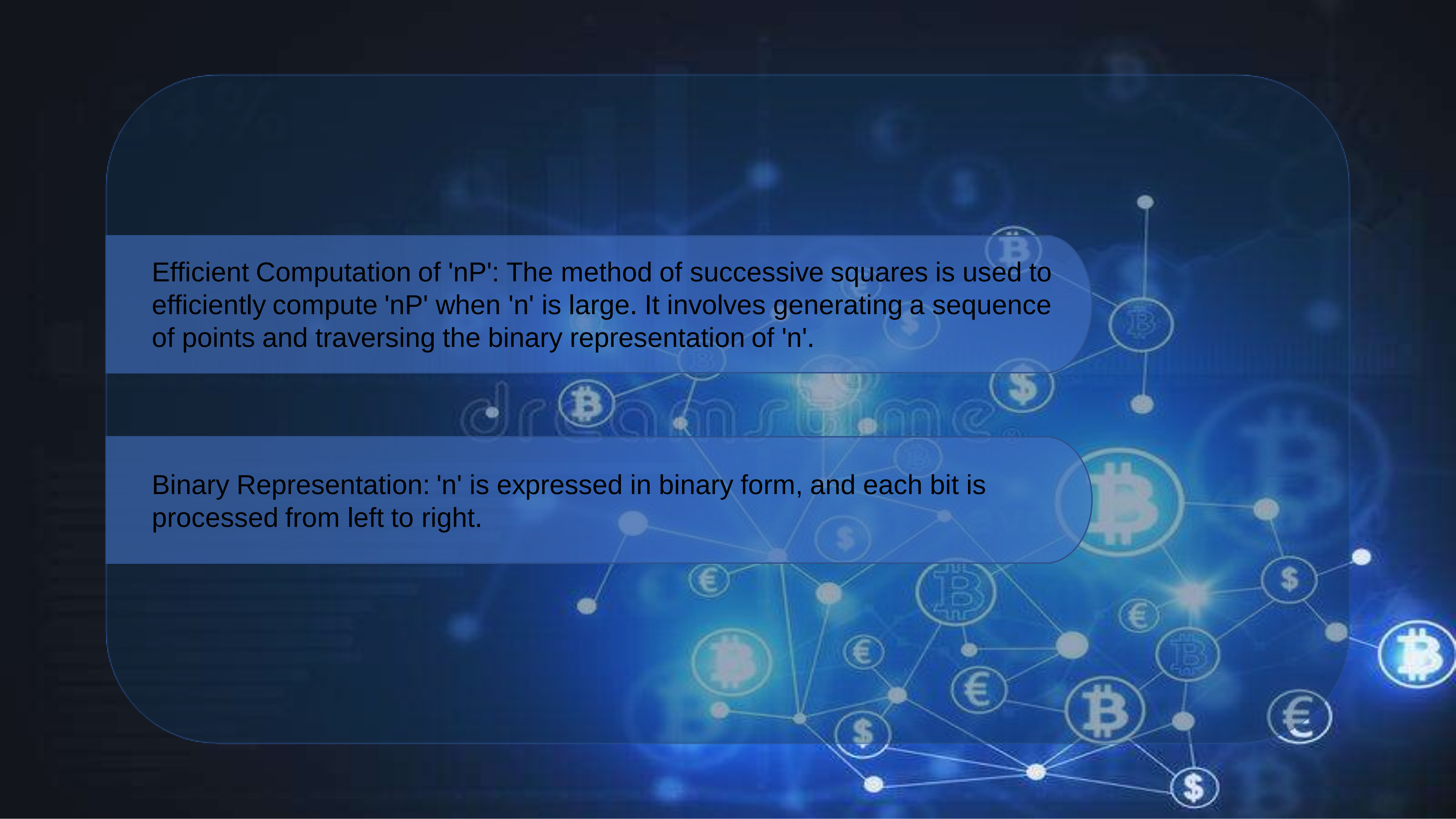


Discrete Logarithm Problem: In Bitcoin and blockchain, the cryptographic algorithm relies on the security of the discrete logarithm problem for elliptic curves over finite fields

Elliptic Curve and Finite Field: The problem involves determining an integer value 'n' such that 'nP = Q' for given points 'P' and 'Q' on an elliptic curve 'c' defined over a finite field ‘F_p'.

Complexity and Simplicity: Computing 'Q = nP' given 'n' and 'P' is efficient, while finding the discrete logarithm 'n' given 'P' and 'Q' is complex, especially for large 'n’.

Advantages of Elliptic Curves: Using elliptic curves for cryptographic algorithms allows for shorter cryptographic keys while maintaining a high level of security, making them suitable for applications like Bitcoin and blockchain.

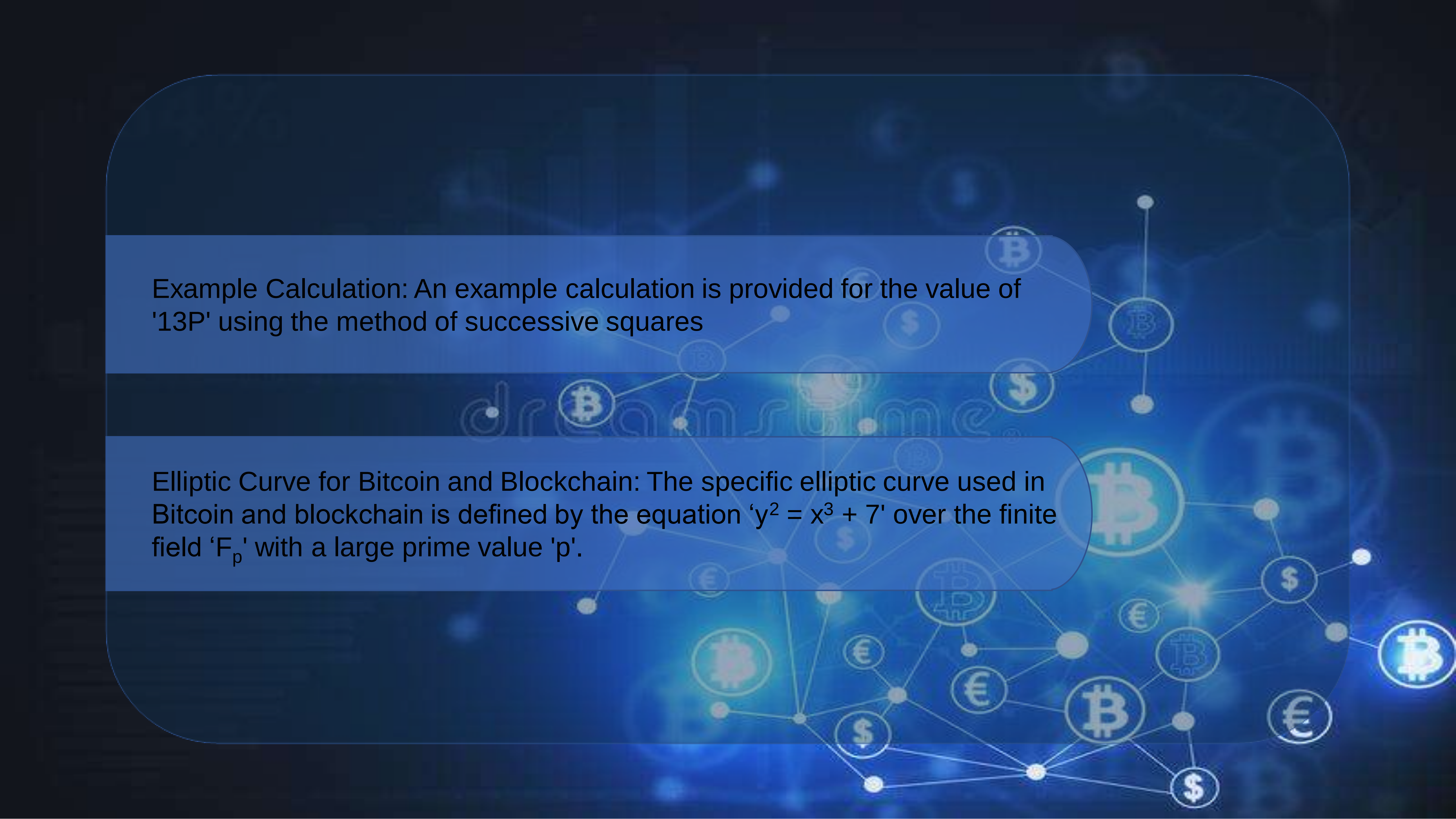


Efficient Computation of 'nP': The method of successive squares is used to efficiently compute 'nP' when 'n' is large. It involves generating a sequence of points and traversing the binary representation of 'n'.

Binary Representation: 'n' is expressed in binary form, and each bit is processed from left to right.

Doubling and Addition: Depending on the value of the current bit, the current point is either doubled or doubled and then added with 'P'.

Binary Representation Summation: The computation of 'nP' involves summing the points based on the binary representation of 'n' and the generated sequence of points.



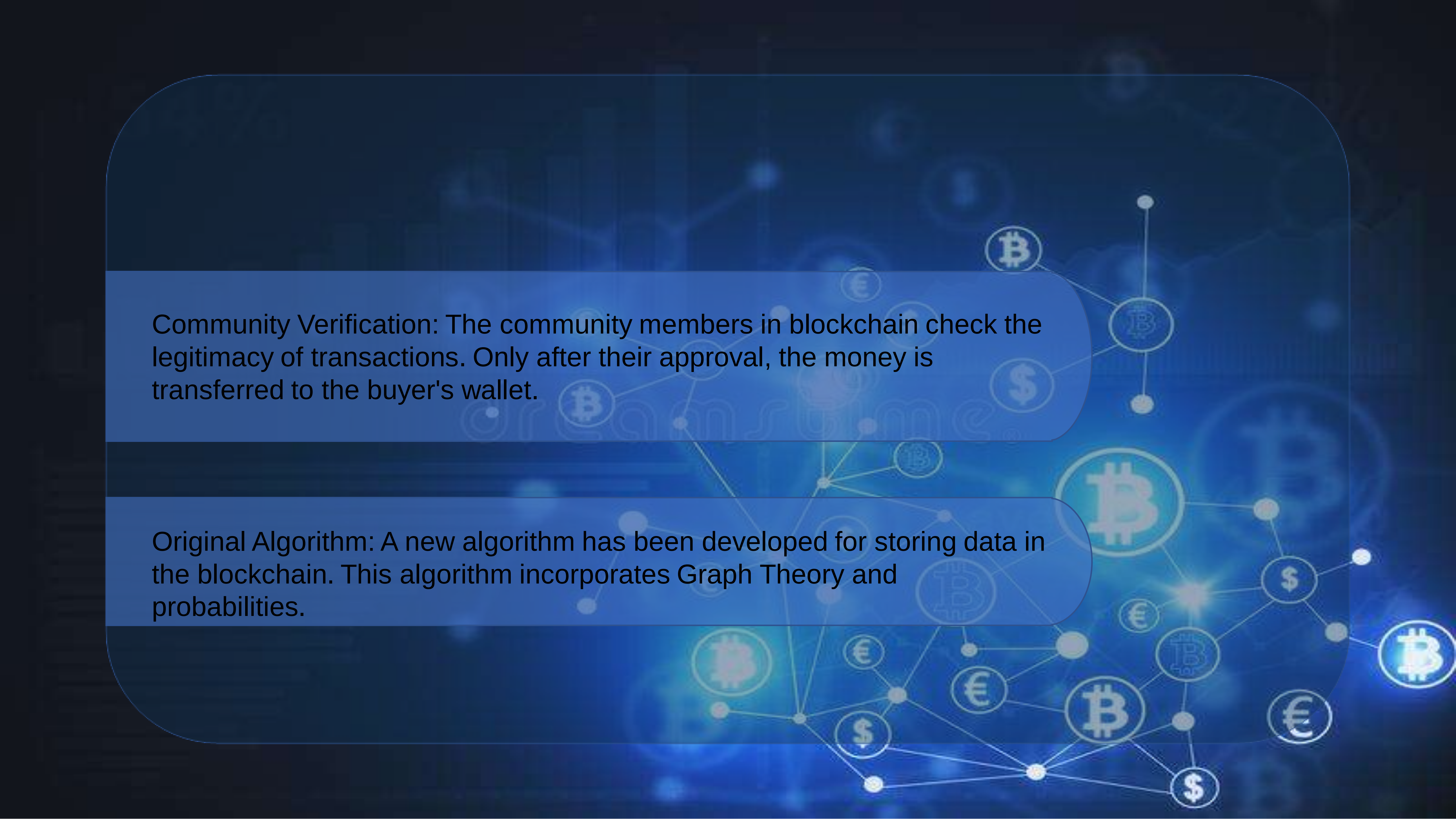
Example Calculation: An example calculation is provided for the value of '13P' using the method of successive squares

Elliptic Curve for Bitcoin and Blockchain: The specific elliptic curve used in Bitcoin and blockchain is defined by the equation ‘$y^2 = x^3 + 7$' over the finite field ‘F_p' with a large prime value 'p'.

ALGORITHMS

Secure Data Algorithm: The commonly used algorithm for securing data is SHA-256, which is a hashing algorithm.

Three Parties in Blockchain: In contrast to traditional banking systems with two parties (buyer and seller), blockchain involves three parties: buyer, seller, and a community of individuals who verify transactions.



Community Verification: The community members in blockchain check the legitimacy of transactions. Only after their approval, the money is transferred to the buyer's wallet.

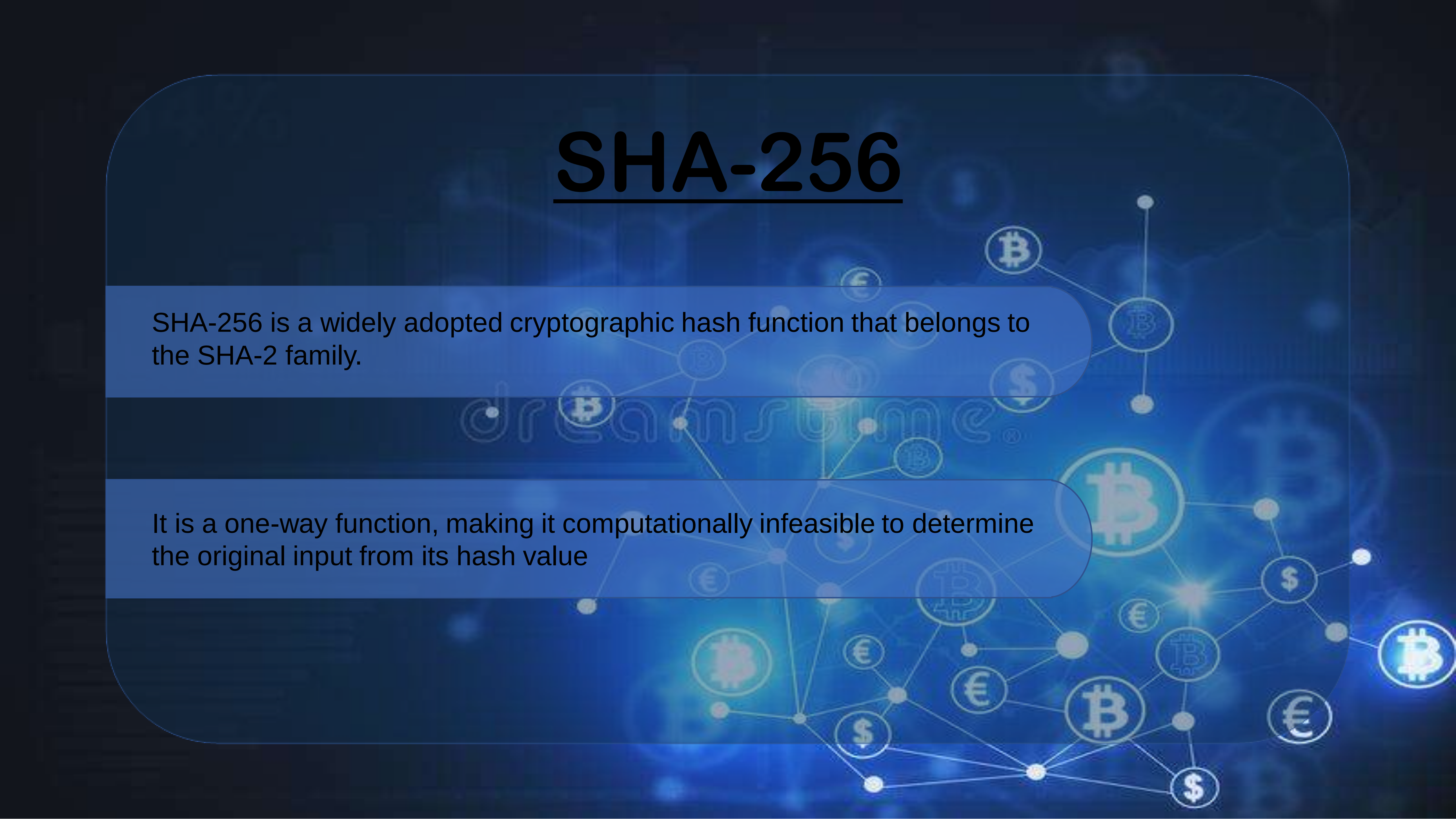
Original Algorithm: A new algorithm has been developed for storing data in the blockchain. This algorithm incorporates Graph Theory and probabilities.

Graph Theory: Graph Theory, a branch of mathematics, is utilized in the new algorithm for data storage in the blockchain.

Probabilities: The algorithm also employs probabilistic methods to enhance the efficiency and reliability of data storage.

The combination of Graph Theory, probabilities, and community verification adds a unique dimension to data storage in blockchain, improving security and transparency in the process.

SHA-256



SHA-256 is a widely adopted cryptographic hash function that belongs to the SHA-2 family.

It is a one-way function, making it computationally infeasible to determine the original input from its hash value

SHA-256 produces a fixed-size output of 256 bits regardless of the input size.

It is designed to be collision-resistant, meaning it is highly improbable to find two different inputs that produce the same hash value.

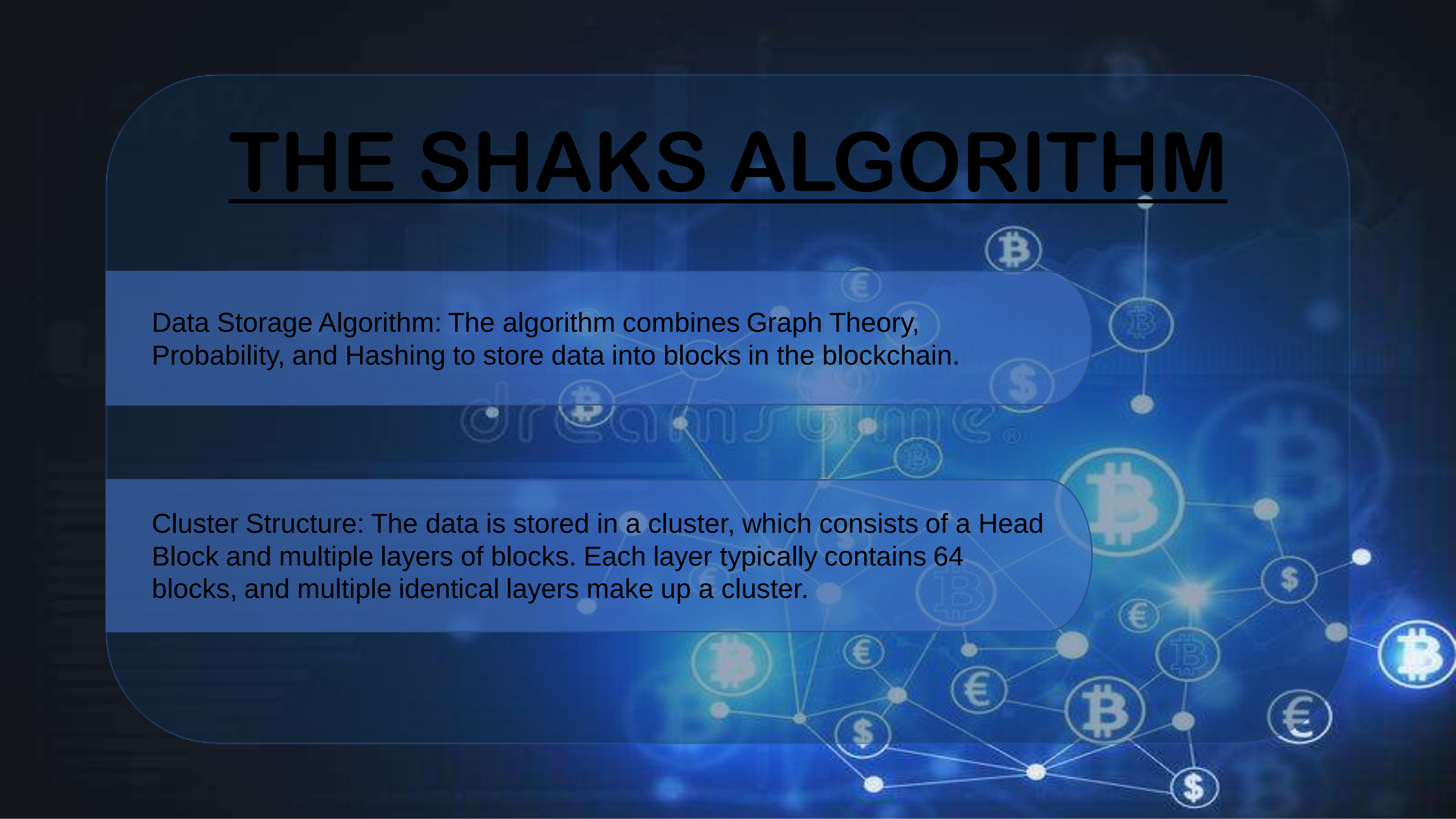
The algorithm is efficient and can process large amounts of data relatively quickly.

HA-256 is considered cryptographically secure and resistant to various known attacks.

It is a standardized algorithm defined in the Federal Information Processing Standards (FIPS) publications by NIST.

SHA-256 provides a means to verify data integrity and detect tampering

Limitations of SHA-256 include vulnerability to quantum computing, collision probability, speed limitations with large data sets, deterministic output, limited input size, and dependence on hash value integrity.



THE SHAKS ALGORITHM

Data Storage Algorithm: The algorithm combines Graph Theory, Probability, and Hashing to store data into blocks in the blockchain.

Cluster Structure: The data is stored in a cluster, which consists of a Head Block and multiple layers of blocks. Each layer typically contains 64 blocks, and multiple identical layers make up a cluster.

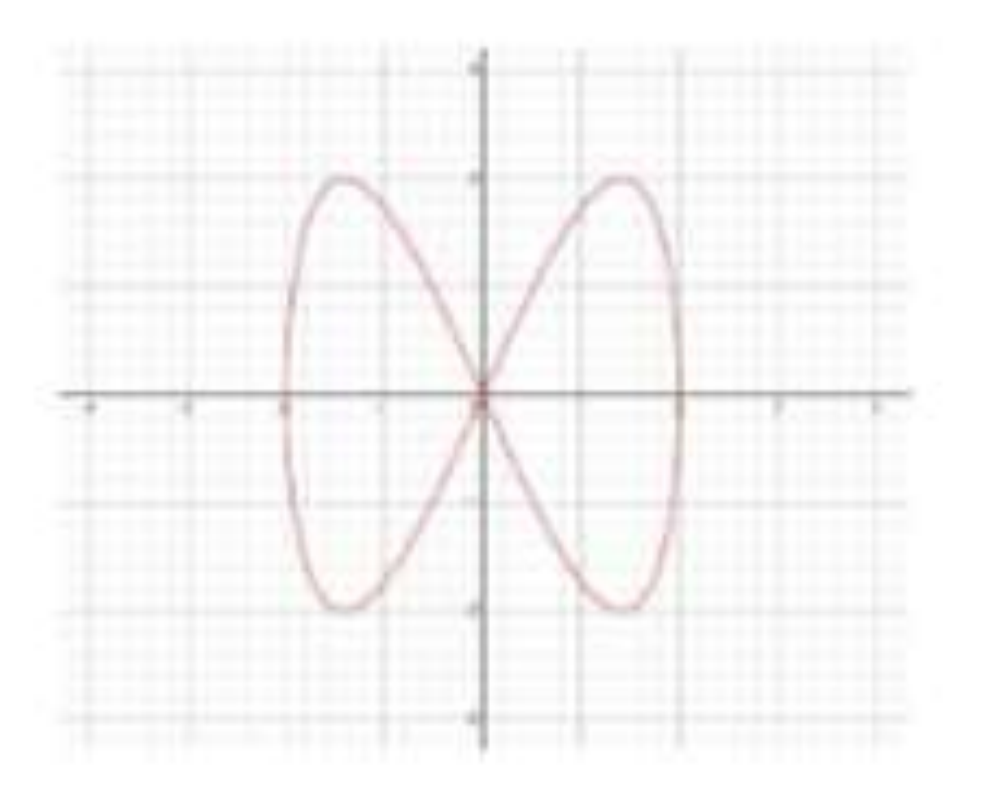
DIAGRAM OF A CLUSTER

Interconnections: All the blocks in a cluster are interconnected, forming an undirected graph. They are also connected to the head node.

Complex Numbers: Complex numbers are utilized in the algorithm, with 'a' representing the public key and 'b' representing the private key. These keys are stored in a complex number, which is made available to all the data.

Hash Function: Data (e.g., passwords or transaction data) is passed through a hash function, converting it into a binary equivalent and performing XOR operations on the digits. The resulting value is then processed to obtain a hash value. The hash function used here is: $y^2-4x^2+x^4=0$

Private Key Multiplication: The private key, accessible only to the blocks within the same cluster, is multiplied with the hash value.



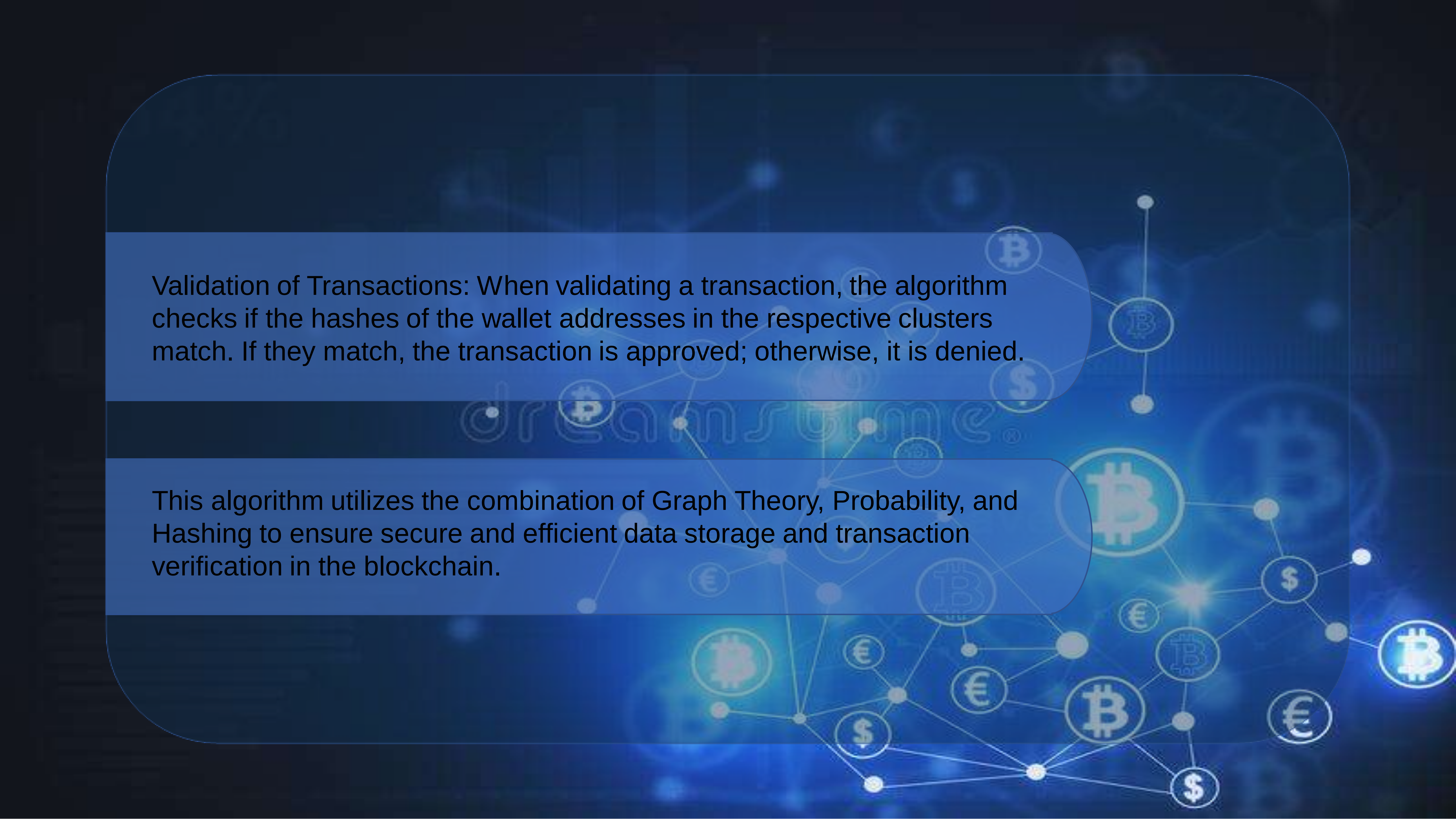
GRAPH OF THE ABOVE FUNCTION

Block Storage: The hash value, multiplied by the private key, is stored in a block. The algorithm navigates through the blocks to find an appropriate block for storage.

Probability and Navigation: Each layer of blocks has a probability associated with it, guiding the navigation process. The probability is passed into the hash function, and the resulting value is added to the existing data.

Block Connection: When a block is full, it disconnects itself from other vertices and remains connected only to the Head Node.

Financial Transaction: In a financial transaction, wallet addresses are stored in different clusters. To access a wallet address, the algorithm enters the Head Node, applies the hash function, multiplies it with the private key, and navigates to the block where the address is stored.



Validation of Transactions: When validating a transaction, the algorithm checks if the hashes of the wallet addresses in the respective clusters match. If they match, the transaction is approved; otherwise, it is denied.

This algorithm utilizes the combination of Graph Theory, Probability, and Hashing to ensure secure and efficient data storage and transaction verification in the blockchain.

CONCLUSION

We have now come to an end of this paper and so far we have discussed a variety of topics including the mathematics behind cryptocurrencies using elliptic curves, we also tried to develop our own algorithm with the help of graph theory and probability and at last also discussed Smart Contracts and how useful they can be in the Decentralized Finance Domain.