RUTUCOIN DOCUMENTATION

Concept of the project.

What is Elliptic curve cryptography?

Elliptic Curve Cryptography (ECC) is a key-based technique for encrypting data. ECC focuses on pairs of public and private keys for decryption and encryption of web traffic.

[Elliptic curve cryptography (ECC)](https://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

ECC bases its approach to public key cryptographic systems on how elliptic curves are structured algebraically over finite fields. Therefore, ECC creates keys that are more difficult, mathematically, to crack. For this reason, ECC is considered to be the next generation implementation of public key cryptography and more secure than RSA.

An elliptic curve for current ECC purposes is a plane curve over a finite field which is made up of the points satisfying the equation:  
y²=x³ + ax + b.

ECC can achieve the same level of security with a 164-bit key that other systems require a 1,024-bit key. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications.

ECC, an alternative technique to RSA, is a powerful cryptography approach. It generates security between key pairs for public key encryption by using the mathematics of elliptic curves.

Elliptic curve cryptography is now used in a [variety of applications](https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/): the U.S. government uses it to protect internal communications, the Tor project uses it to help assure anonymity, it is the mechanism used to [prove ownership of bitcoins](https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm), it provides signatures in [Apple's iMessage service](https://blog.quarkslab.com/imessage-privacy.html), it is used to encrypt DNS information with [DNS Curve](https://en.wikipedia.org/wiki/DNSCurve), and it is the preferred method for authentication for secure web browsing over SSL/TLS.

What is SHA-256 Cryptographic Hash Algorithm?

A cryptographic hash (sometimes called ‘digest’) is a kind of ‘signature’ for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text. See [below](https://www.movable-type.co.uk/scripts/sha256.html#src-code) for the source code. A hash is not ‘encryption’ – it cannot be decrypted back to the original text (it is a ‘one-way’ cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare ‘hashed’ versions of texts, as opposed to decrypting the text to obtain the original version.

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long.

Packages used to implement the idea

1. Bn.js – <https://www.npmjs.com/package/bn.js?activeTab=readme>
2. Crypto.js – <https://www.npmjs.com/package/crypto-js>
3. Elliptic.js – <https://www.npmjs.com/package/elliptic>
4. Node.js(crypto) - <https://nodejs.org/api/crypto.html>