# Node.js: cifrare il numero di una carta di credito

Con Node.js possiamo cifrare il numero di una carta di credito.

Creiamo prima una chiave casuale e quindi utilizziamo l'algoritmo di cifratura AES-256.

```
'use strict';

const crypto = require('crypto');
const CIPHER_ALGORITHM = 'aes-256-ctr';

const createKey = () = > {
    let str =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789$%&/()=?^"!|[]{}*+-:.;,_@#<>';
    return str.split('').sort((a, b) => {return
Math.random() - 0.5}).join('');
};

const key = createKey();

class KeyGen {
    constructor(key, algorithm) {
        this.key = key;
        this.algorithm = algorithm;
    }

    cypher(str) {
        let sha256 = crypto.createHash('sha256');
```

```
        sha256.update(this.key);
        let iv = crypto.randomBytes(16);
        let cipher =
crypto.createCipheriv(this.algorithm,
sha256.digest(), iv);
        let ciphertext =
cipher.update(Buffer.from(str));
        let  encrypted = Buffer.concat([iv,
ciphertext, cipher.final()]).toString('base64');
        return encrypted;
    }

    decypher(enc) {
        let sha256 = crypto.createHash('sha256');
        sha256.update(this.key);
        let input = Buffer.from(enc, 'base64');
        let iv = input.slice(0, 16);
        let decipher =
crypto.createDecipheriv(this.algorithm,
sha256.digest(), iv);
        let ciphertext = input.slice(16);
        let plaintext = decipher.update(ciphertext) +
decipher.final();
        return plaintext;
    }
}

let kg = new KeyGen(key, CIPHER_ALGORITHM);
let enc = kg.cypher('4111111111111111');
console.log(enc); //
'F6NR6AeK475VsnH874uj2P9bxRCk8mO14gWqDXpAg5o='
console.log(kg.decypher(enc)); // '4111111111111111'
```

Attenzione: la cifratura da sola non è sufficiente a garantire la sicurezza dei dati secondo gli standard correnti. Occorre munirsi di un'infrastruttura hardware e software tale da permettere la rotazione ciclica delle chiavi di cifratura e la protezione sugli accessi all'infrastruttura stessa.

# Riferimenti

PCI Security