Cifrado de sockets

Los sockets se pueden cifrar en NodeJS. Se pueden usar cifrados TLS o SSL.

Para cifrar un socket se necesitan los certificados públicos y privados. Para ello se puede usar el comando:

```
 $ openssl req -nodes -new -x509 -keyout server.key -out server.cert
```

Se generarán dos certificados “server.key” y “server.cert”.

Con estos certificados se puede cifrar el socket de un servidor:

```
'use strict';

const tls = require('tls');
const fs = require('fs');
const port = 8000;

const options = {
  key: fs.readFileSync('server.key'),
  cert: fs.readFileSync('server.cert'),
  //ca: fs.readFileSync('ca.crt'), // authority chain for the clients
  requestCert: false, // Preguntar por el certificado del cliente
  rejectUnauthorized: false, // Evitar el acceso de clientes sin autorizar
};

var server = tls.createServer(options, (socket) => {
  socket.setEncoding('utf8');
  socket.write(JSON.stringify({'saludo': 'Hola', 'contador': 0}));
  socket.on('data', (data) => {
    console.log(data);
    const a = JSON.parse(data);
    a.contador = parseInt(a.contador) + 1;
    socket.write(JSON.stringify(a));
  })
});

server.on('connection', function(c)
  {
    console.log('insecure connection');
  })

server.on('secureConnection', function (c)
  {
    // c.authorized es true si el cliente presenta un certificado que se pueda
validar con el ca
    console.log('secure connection; client authorized: ', c.authorized);
  })


server.listen(port, function() {
  console.log('server listening on port ' + port + '\n');
});
```

Para cifrar el cliente también se le generarán unos certificados “client.key” y “client.cert”. Usando estos certificados se puede cifrar la conexión del cliente:

```
'use strict';
```

```
const port = 8000;
const hostname = 'localhost';

const tls = require('tls');
var fs = require('fs');

const options = {
  host: hostname,
  port: port,

  // Claves del cliente:
  key: fs.readFileSync('client.key'),
  cert: fs.readFileSync('client.cert'),

  // Sólo se usa si el servidor usa certificados autofirmados:
  //ca: fs.readFileSync('ca.crt')
  requestCert: false, // Preguntar por el certificado del cliente
  rejectUnauthorized: false, // Evitar el acceso de clientes sin autorización
};

var socket = tls.connect(options, () => {
  console.log('client connected', socket.authorized ? 'authorized' :
'unauthorized');
  console.log('Tipo de cifrado:');
  console.log(socket.getCipher());
  socket.write(JSON.stringify({'contador': 0}));
});

socket.setEncoding('utf8');

socket.on('data', (data) => {
  console.log(data);
  const a = JSON.parse(data);
  a.contador = parseInt(a.contador) + 1;
  if(a.contador > 10) socket.end();
  else socket.write(JSON.stringify(a));
});

socket.on('end', () => {
  console.log("End connection");
});
```