Основы и принципы криптографии

JavaScript Courses

www.courses.dp.ua

Основы криптографии

1. Двоичная и шестнадцатеричная система счисления;
2. Битовые операции AND, OR, XOR;
3. Симметричное шифрование;
4. Ассиметричное шифрование;
5. Хеширование;
6. Цифровая подпись;
7. HTTPS;

Двоичная и шестнадцатеричная система счисления

```
BIN: 0; DEC: 0; HEX: 0;
BIN: 1; DEC: 1; HEX: 1;
BIN: 10; DEC: 2; HEX: 2;
BIN: 11; DEC: 3; HEX: 3;
BIN: 100; DEC: 4; HEX: 4;
BIN: 101; DEC: 5; HEX: 5;
BIN: 110; DEC: 6; HEX: 6;
BIN: 111; DEC: 7; HEX: 7;
BIN: 1000; DEC: 8; HEX: 8;
BIN: 1001; DEC: 9; HEX: 9;
BIN: 1010; DEC: 10; HEX: a;
BIN: 1011; DEC: 11; HEX: b;
BIN: 1100; DEC: 12; HEX: c;
BIN: 1101; DEC: 13; HEX: d;
BIN: 1110; DEC: 14; HEX: e;
BIN: 1111; DEC: 15; HEX: f;
BIN: 10000; DEC: 16; HEX: 10;
BIN: 10001; DEC: 17; HEX: 11;
BIN: 10010; DEC: 18; HEX: 12;
BIN: 10011; DEC: 19; HEX: 13;
BIN: 10100; DEC: 20; HEX: 14;
```

Разрядность системы счисления зависит от количества цифр используемых для формирования чисел, в остальном отличий от привычной нам десятичной системы нет.

Битовые операции

		AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

```
var a = 123;
var b = 77;

var c = a & b; // Битовый оператор И
var d = a | b; // Битовый оператор ИЛИ
var e = a ^ b; // Битовый оператор XOR (исключающее или)

console.log(c, d, e);
```

```
73 127 54
```

Битовые операторы выполняют операции над битами числа

https://learn.javascript.ru/bitwise-operators

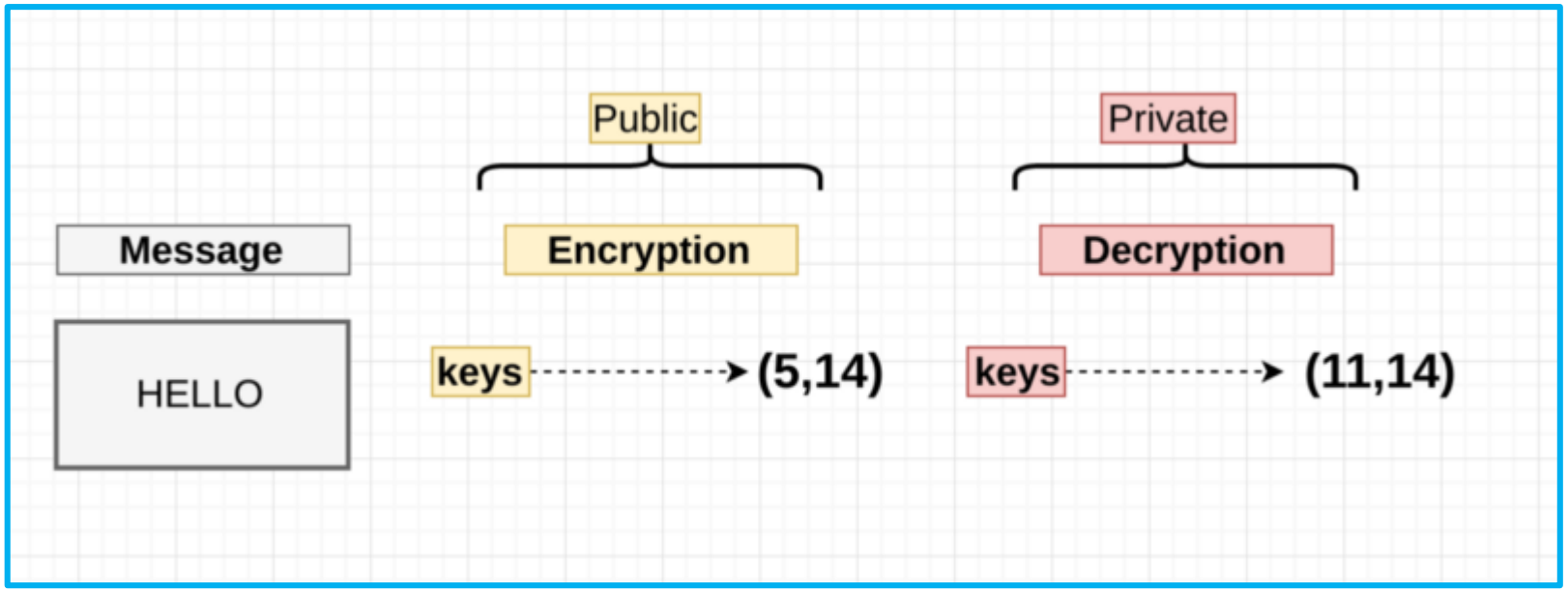
```
2
3   let data = "Hello world!";
4
5   let dataCodes = [...data].map(item => item.charCodeAt(0));
6
7   console.log(dataCodes);
8
9   let key = 123; // Secret Key
10
11  // Encoding
12  let encoded = dataCodes.map( item => item ^ key );
13
14  console.log(encoded);
15
16  let encodedData = String.fromCharCode(...encoded);
17
18  console.log(encodedData);
19
20  //Decoding
21  let decoded = encoded.map( item => item ^ key);
22
23  console.log(decoded);
24
25  let decodedData = String.fromCharCode(...decoded);
26
27  console.log(decodedData);
```

Симметричное шифрование

Симметричное шифрование – использует один и тот же ключ для шифрования и расшифровки данных

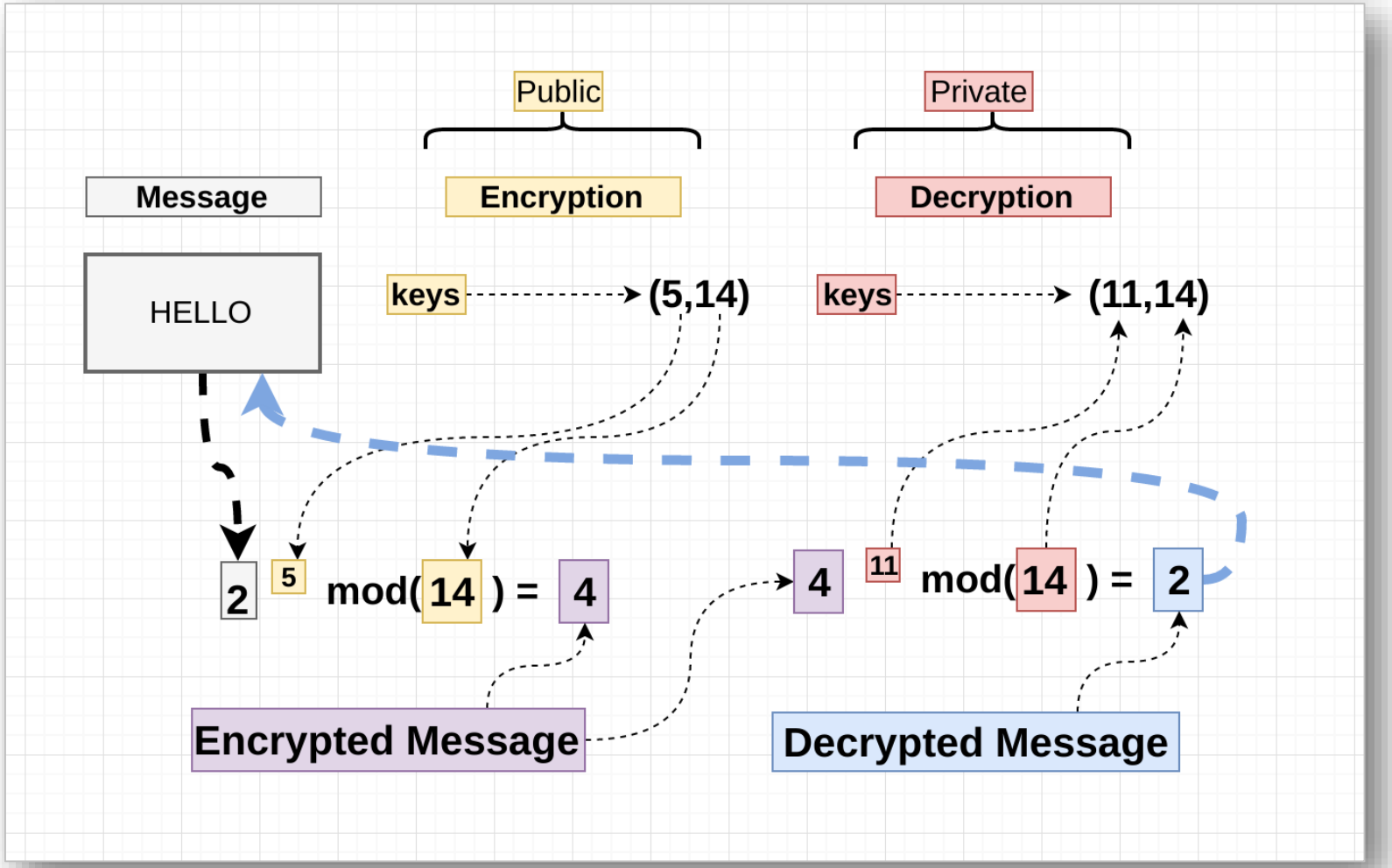
https://ru.wikipedia.org/wiki/Симметричные_криптосистемы

Ассиметричное шифрование (алгоритм RSA)



Ассиметричное шифрование – использует разные ключи (**открытый** и **закрытый**) для шифрования и расшифровки данных.

https://ru.wikipedia.org/wiki/Криптосистема_с_открытым_ключом



Ассиметричное шифрование (алгоритм RSA)

Ассиметричное шифрование – использует разные ключи (открытый и закрытый) для шифрования и расшифровки данных.

Алгоритм RSA / Генерация ключей

1. Выбираем два простых числа **P** и **Q**;
2. Находим **N = P * Q**;
3. Находим **F = (Q – 1) * (P – 1)**;
4. Подбираем число **E**, которое должно быть простым, быть меньшим **F** и их максимальный общий делитель был **1**;
5. Выбираем число **D** удовлетворяющее **D * E % F == 1**;
6. Теперь у нас есть пара ключей **(E, N)** и **(D, N)**;

Node-RSA

В **NPM’**е есть модуль **node-rsa** содержащий всё необходимо для шифрования/дешифрования по алгоритму **RSA**.

https://www.npmjs.com/package/node-rsa

Хеширование

Hello world!!**!** => **(SHA256)** => 4354dfda70c8f0d3991b9de3d56dcb6e9f2fc6c0316d235b63afeb388471ada4

Hello world!! => **(SHA256)** => bbca77170621e018f9b8d17c850d2c7efe3cf9998cf741edf8e7dffbaeeb160e

Хеширование по алгоритму SHA256 (калькулятор):
http://www.xorbin.com/tools/sha256-hash-calculator

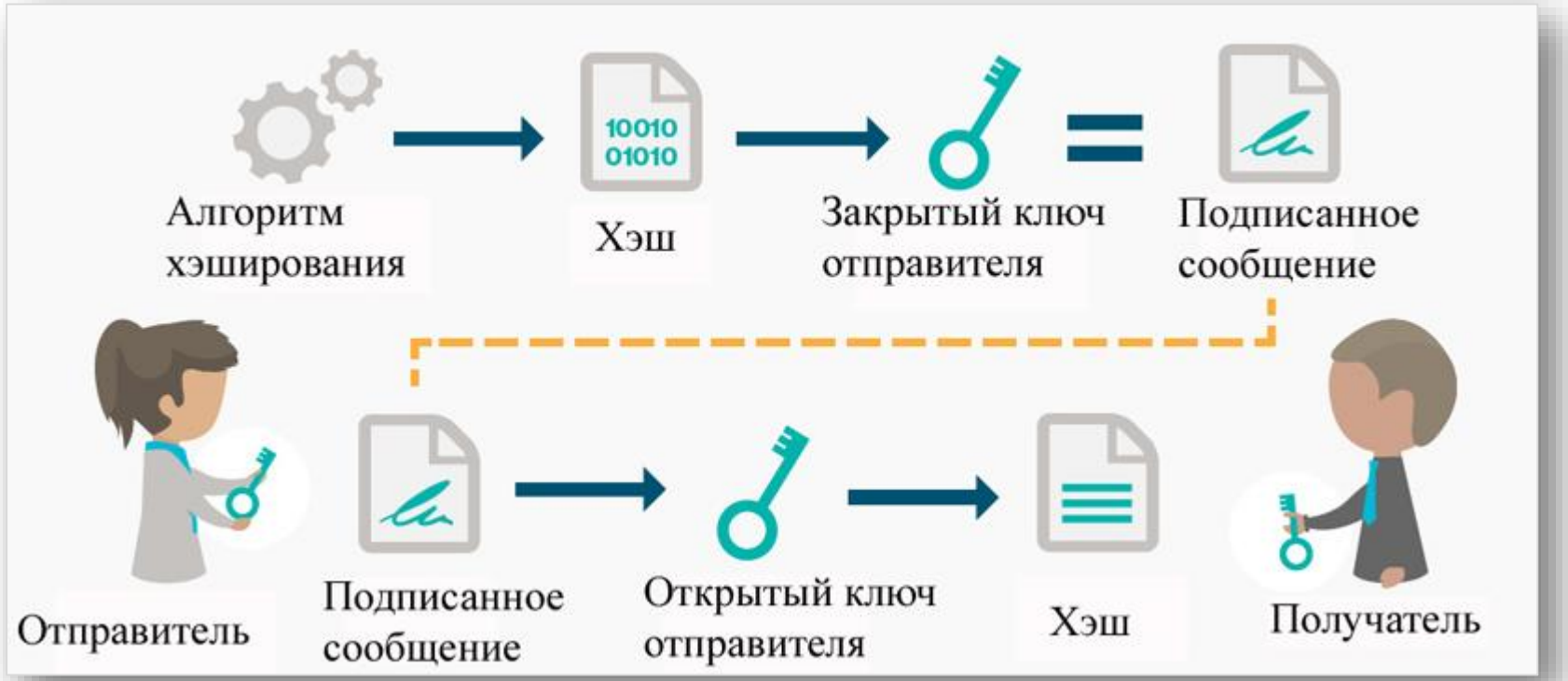
Преобразование входного набора данных любого (как правило большого) размера в данные фиксированного размера.
Существует множество алгоритмов хеширования.

https://ru.wikipedia.org/wiki/Хеширование

*В **NPM'е** есть модуль **sha256** функцию выполняющую расчёт хеша по указанному алгоритму.*

https://www.npmjs.com/package/js-sha256

Цифровая подпись



Цифровая подпись *– технология на базе хеширования и ассиметричного шифрования задача которой подтвердить достоверность передаваемых данных от отправителя к получателю.*

https://ru.wikipedia.org/wiki/Электронная_подпись

HTTPS

HTTP с шифрованием, версия протокола обеспечивающая шифрование (ассиметричное) всех передаваемых данных между браузером и веб-сервером. Базируется на технологии цирковой подписи.

https://ru.wikipedia.org/wiki/HTTPS