Контроль целостности данных

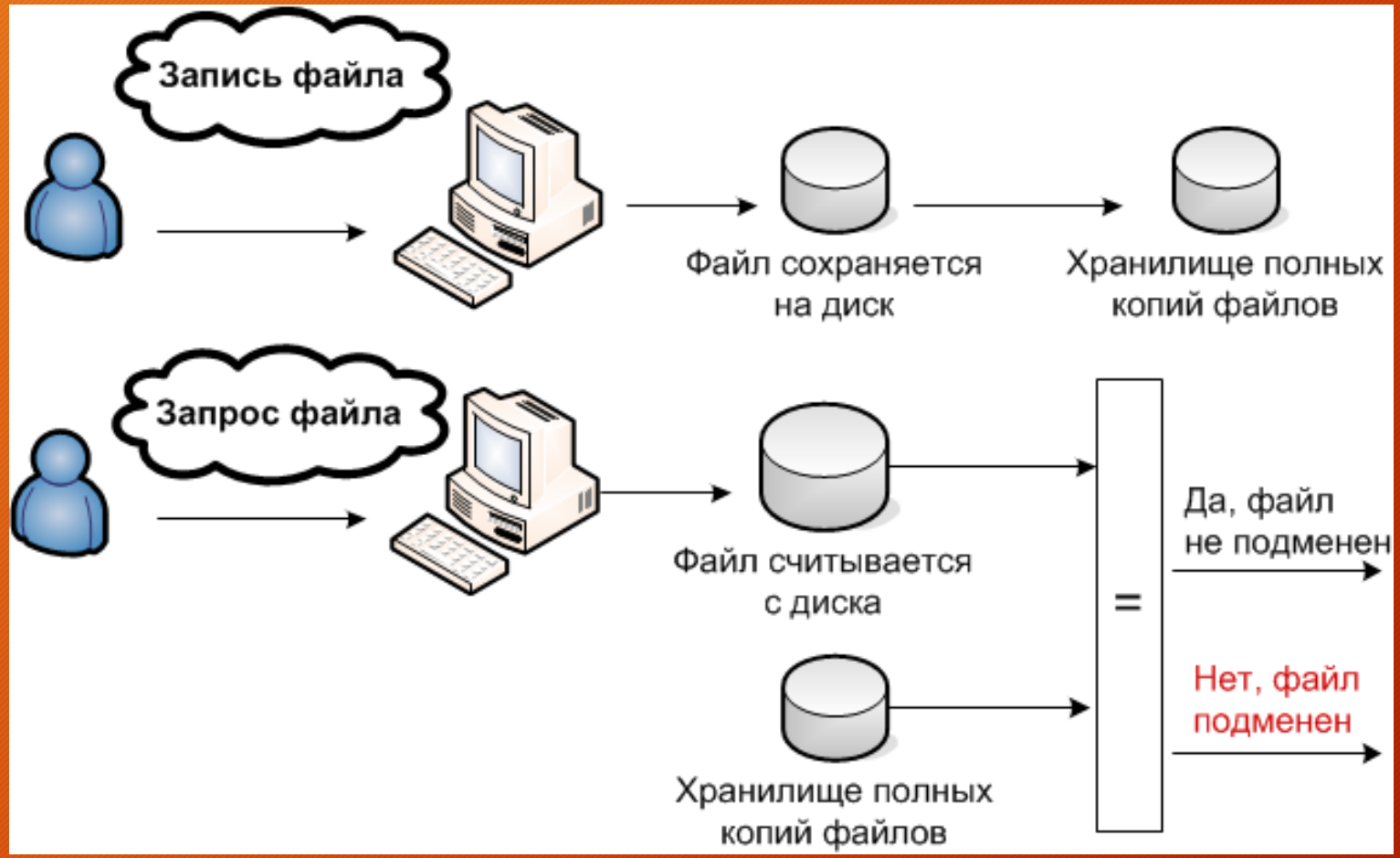
Контроль целостности данных

Методы контроля целостности данных:

- Полная копия данных
- Контрольная сумма
- Хеш
- Имитовставка
- ЭЦП

Целостность данных (в широком смысле) — это состояние информации, при котором отсутствует любое её изменение либо изменение осуществляется только преднамеренно субъектами, имеющими на него право.

Полная копия данных



Преимущества:

- простота реализации
- полный контроль данных (до бита)

Недостатки:

- большой объем
- копии можно подменить
- копиями можно воспользоваться (например: если данные - пароль)

Контрольная сумма

Контрольная сумма - значение, рассчитанное по входным данным с помощью определённого алгоритма.

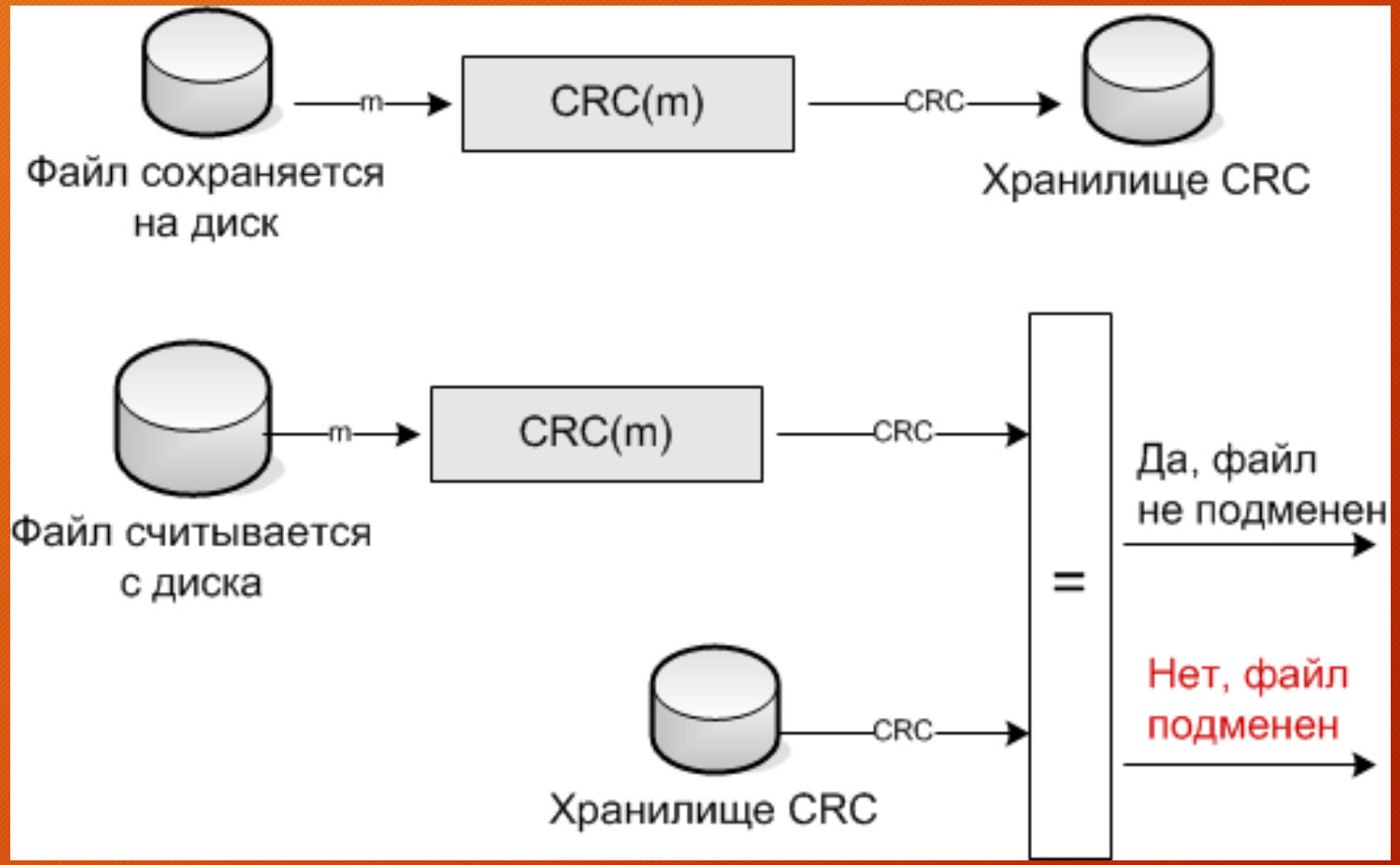
Преимущества:

- высокая скорость вычисления

Недостатки:

- можно подменить
- для одного значения существует множество исходных данных
- можно подобрать исходные данные к значению за приемлемое время (например: получить пароль)

Контрольная сумма



Примеры контрольных сумм: CRC8, CRC16, CRC32

Пример вычисления:

исходный текст: Контроль целостности данных

crc32 (длина 32 бита)

b4feb5a5

Хеш

Хеш (хэш, криптографический хеш) - значение, рассчитанное по входным данным с помощью криптографического алгоритма.

Преимущества:

- нельзя подобрать исходные данные к значению за приемлемое время (например: получить пароль)

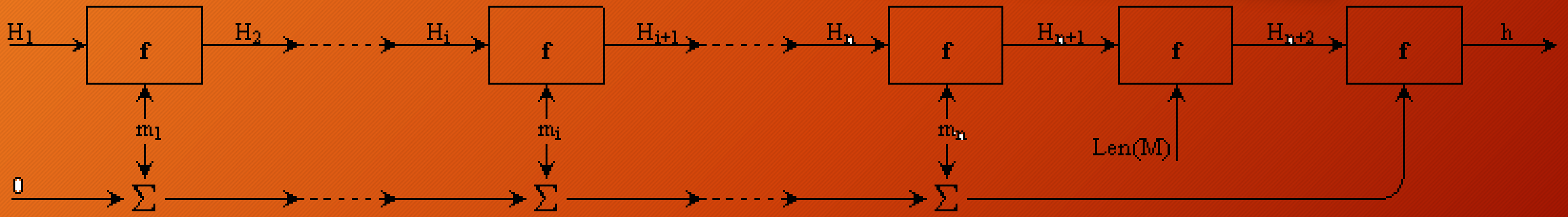
Недостатки:

- низкая скорость вычисления (сопоставима с шифрованием)
- можно подменить
- для одного значения существует множество исходных данных

Хеш

Вычисляют хеш шифрованием данных блочным алгоритмом в режимах CBC, но со стандартным (известным) ключом. Хешем является последний шифрованный блок.

ГОСТ Р 34.11-94



Входное сообщение M разделяется на блоки mn,mn - 1,mn ?-2,...,m1 по 256 бит. В случае если размер последнего блока mn меньше 256 бит, то к нему приписываются слева нули для достижения заданной длины блока.

Каждый блок сообщения подаётся на шаговую функцию для вычисления промежуточного значения хеш-функции Hout=f(Hin, mi) где Hout, Hin, mi — блоки длины 256 бит.

Пример вычисления

исходный текст: Контроль целостности данных

md2 (длина 128 бит)
7e347a5f3a3d8c6837d7accbed3e0b1e
md4 (длина 128 бит)
e37e491b9b4ea133df9964b25d7c7cd9
md5 (длина 128 бит)
8ab8a5cf989e220ff8d39be415b903d5
sha1 (длина 160 бит)
63cdc45bd8a857007d0be8c435e1e547653482d3
sha224 (длина 224 бит)
670c98e5b7ce7bd2c7efb98b6ed448e0a6f3247e3
fdeb678dde61bce
sha256 (длина 256 бит)
1a97bcc0fbcf6722a8aac7b73d12700c00227789
04790ce9eee7656f12d95dc2
sha384 (длина 384 бит)
b87b59cad0db141378d8ff04e46d00dd1371133
91d2a7009d640dd018680c459
310bfbdfef6b3258aeaa4b8146105698
sha512 (длина 512 бит)
3f7b9d8b24fb1d764026fe2b0f72ad62d65fd1c5
d77a18784108a69e8ad172c2
e6583989439aea658a2111525897d43af0f6c50c
f299b360c21b3e02e8706f4e
ГОСТ Р 34.11-94 (длина 256 бит)
d38e4f1bc5d03601486f4aca83fed00c82e1a36f
dac27806cce4b9464af1e9f9

ГОСТ Р 34.11-94

Применение:

контроль целостности файлов
контроль передаваемых данных по каналам связи
контроль целостности при считывании данных (например: с HDD)
хеши паролей
аутентификация (CRAM-MD5, DIGEST-MD5 и т.д.)

Имитовставка (MAC, message authentication code — код аутентичности сообщения)

Имитовставка - значение, рассчитанное по входным данным с помощью криптографического алгоритма с использованием секретного элемента (ключа), известного только отправителю и получателю.

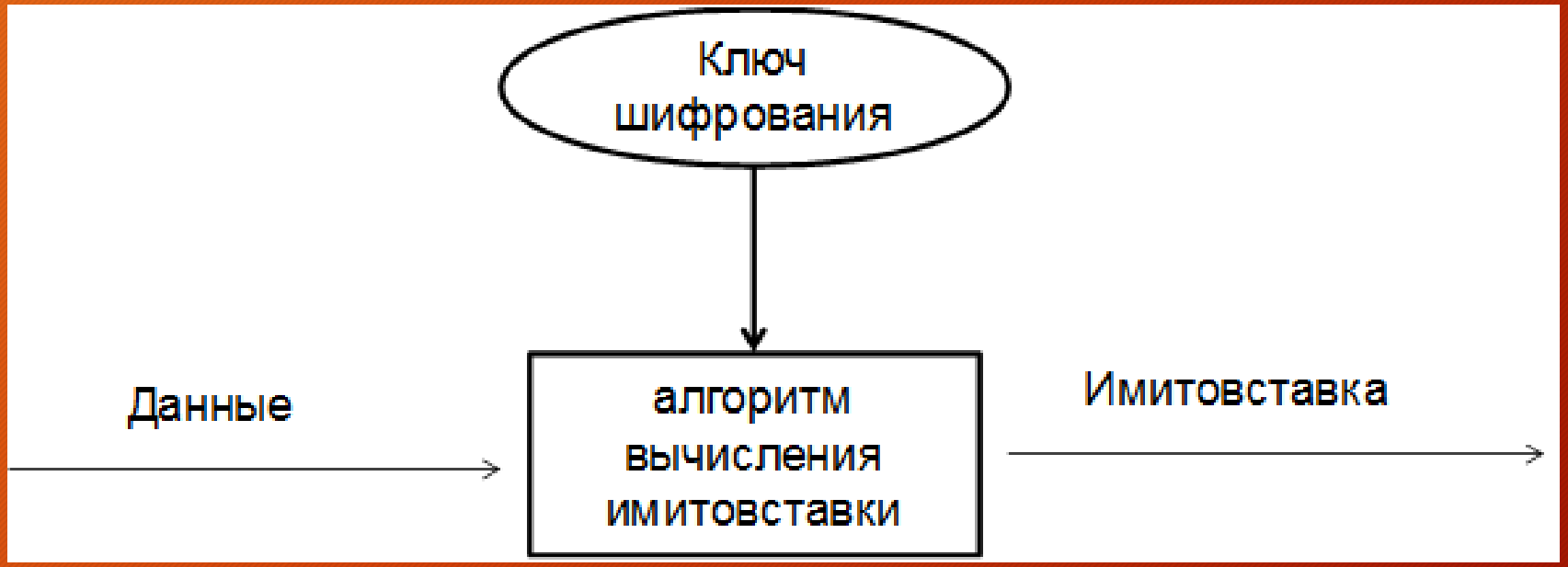
Преимущества:

- нельзя подобрать исходные данные к значению за приемлемое время (например: получить пароль)
- нельзя подменить без секретного элемента (ключа)

Недостатки:

- низкая скорость вычисления (сопоставима с шифрованием)
- для одного значения существует множество исходных данных
- секретный ключ известен как минимум двоим

Вычисление имитовставки



Имитовставка по ГОСТ 28147-89

Длина имитовставки от 1 до 32 бит.

Открытый текст T_O разбивается на блоки длиной 64 бита. Последний блок в случае необходимости дополняется нулями.

$$T_O = T_O^{(1)} T_O^{(2)} \dots T_O^{(N)}$$

Первый блок $T_O^{(1)}$ шифруется, что и сообщение, но с применением 16 циклов вместо 32. Результат по битам по модулю 2 складывается с вторым блоком $T_O^{(2)}$ и так же шифруется. Результат складывается с третьим блоком... и так далее.

$$I = E'_k(T_O^{(N)} \oplus E'_k(T_O^{(N-1)} \oplus E'_k(\dots E'_k(T_O^{(2)} \oplus E'_k(T_O^{(1)})) \dots))$$

Имитовставка по ГОСТ 28147-89

Применение:
контроль целостности файлов
контроль передаваемых данных по каналам связи
аутентификации источника данных (не во всех случаях)
Обычные хэш-алгоритмы использовать для вычисления имитовставки нельзя (MD5 и т.д.) т.к. отсутствует секретный ключ. Поэтому создан HMAC.
HMAC (Hash-based Message Authentication Code) - механизм включения секретного ключа в существующие хэш-алгоритмы.

Электронная цифровая подпись

Электронная цифровая подпись - зашифрованное значение вычисленного хеша по входным данным.

Преимущества:

- нельзя подобрать исходные данные к значению за приемлемое время (например: получить пароль)
- нельзя подменить без секретного элемента (ключа)
- секретный ключ известен одному

Недостатки:

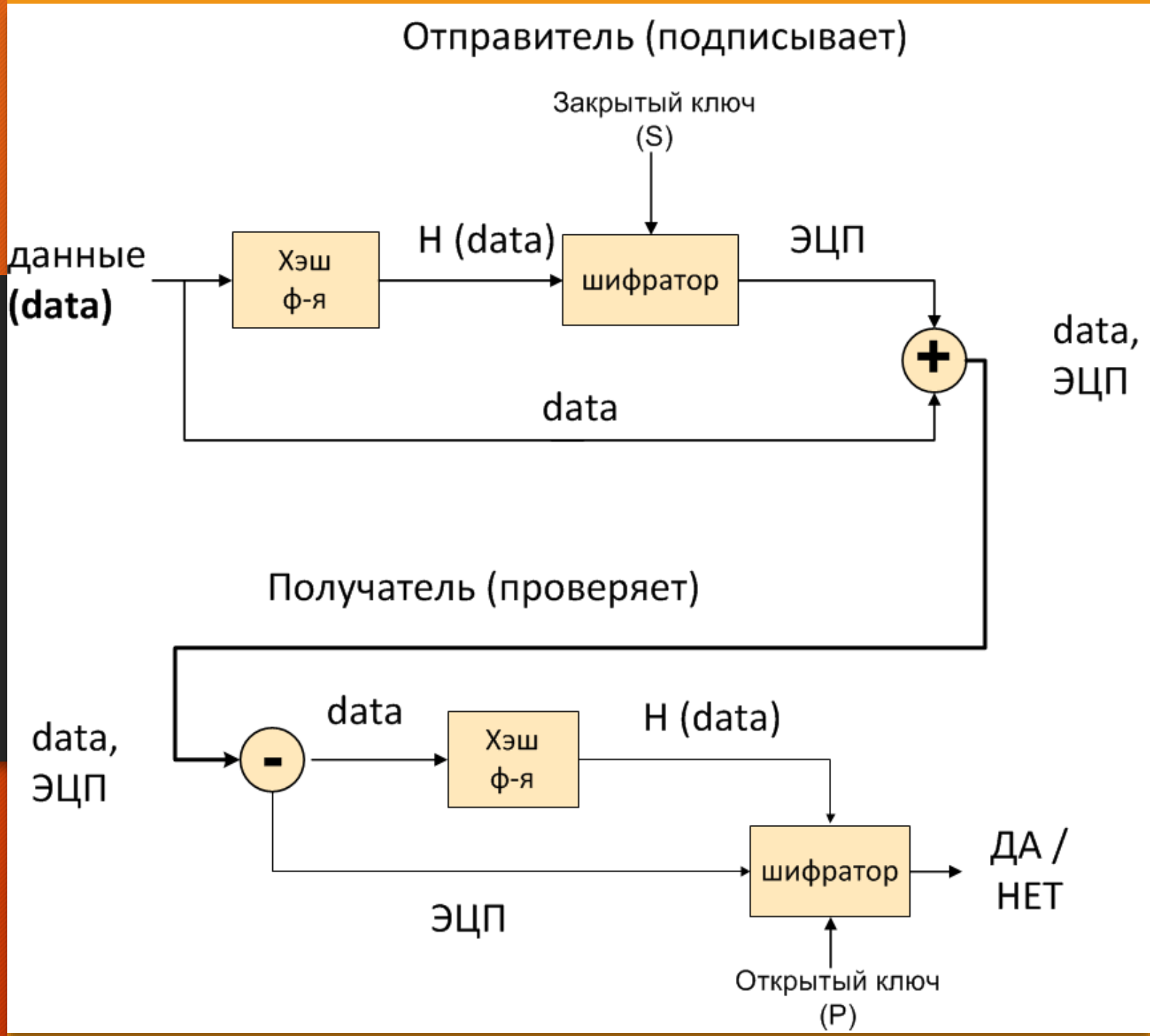
- низкая скорость вычисления (сопоставима с шифрованием)
- для одного значения существует множество исходных данных

Создание и проверка ЭЦП

Алгоритм:

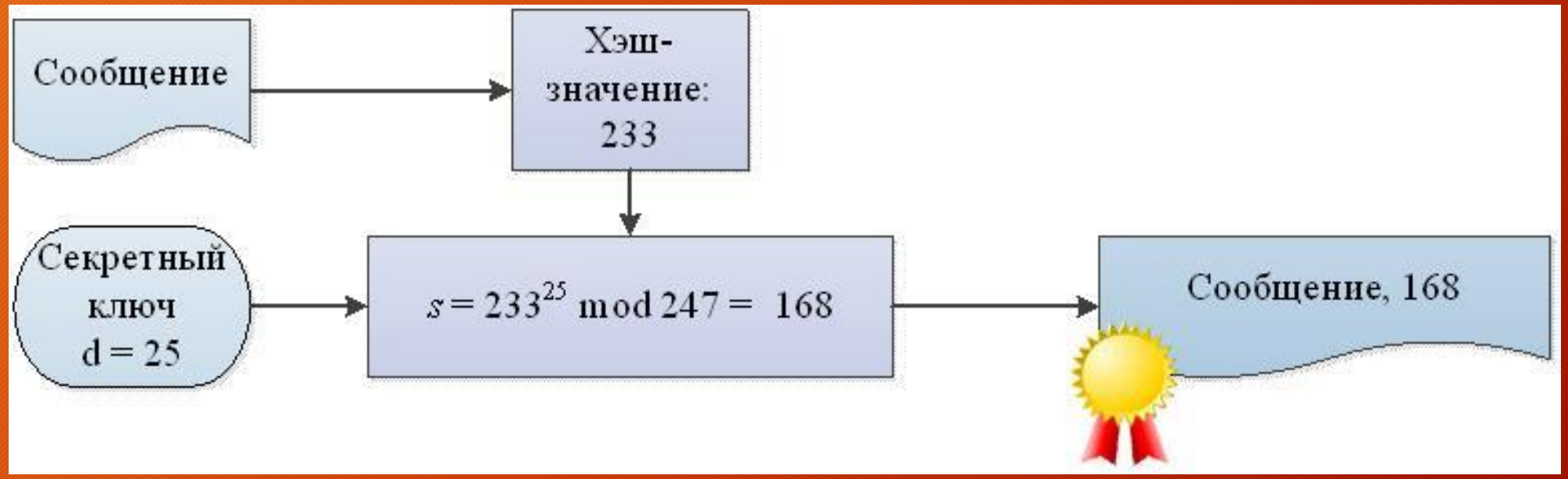
вычисляется хеш

шифруется хеш

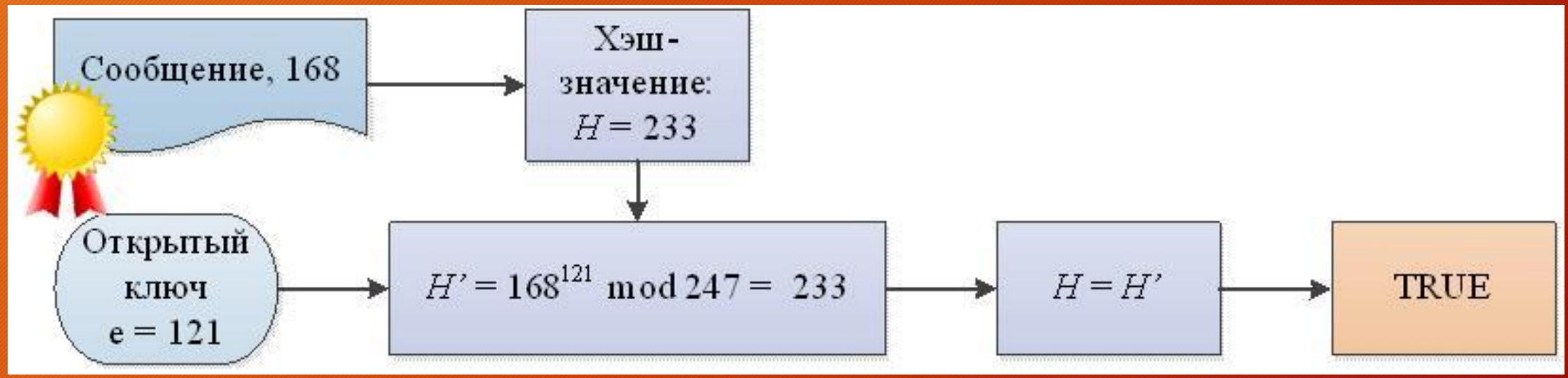


Факторизация больших чисел
Генерация ключей

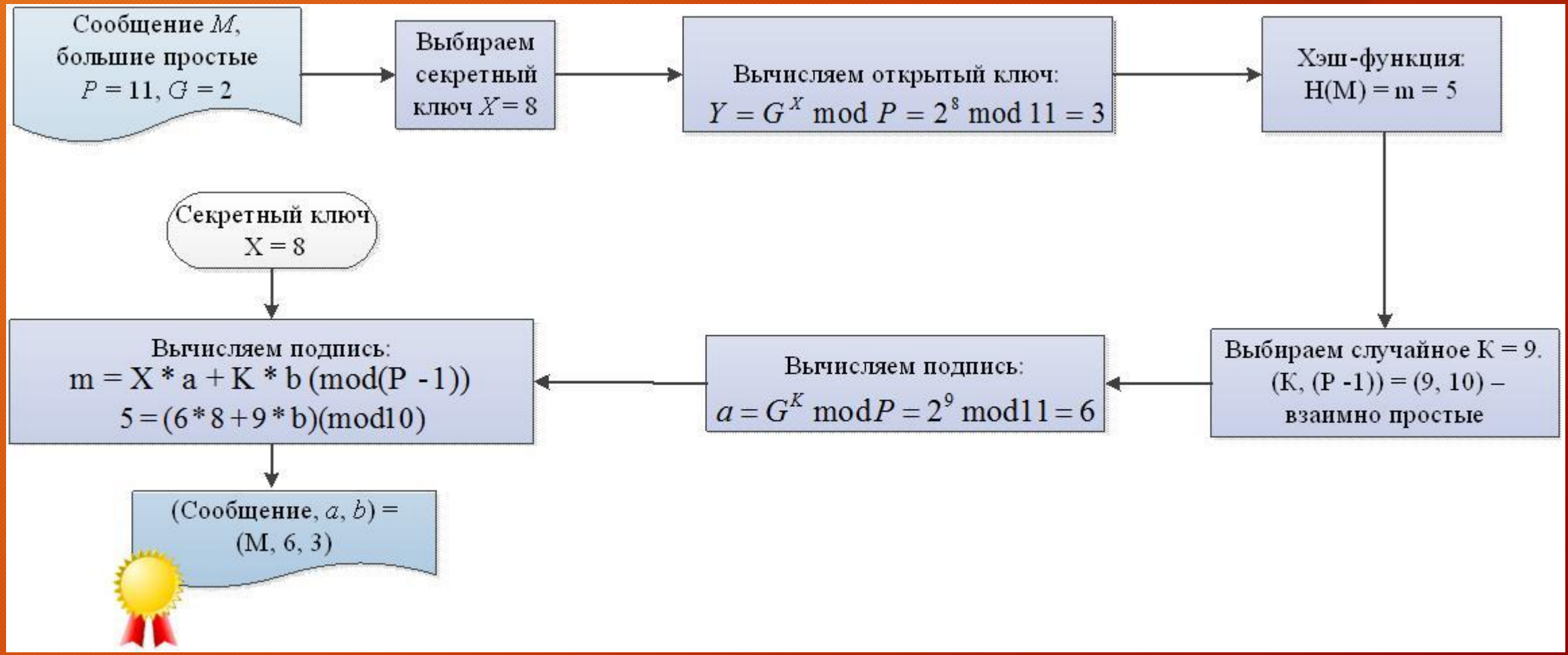
Вычисление электронной подписи



Проверка электронной подписи



Дискретное логарифмирование
Генерация подписи



Проверка электронной подписи