v. 2.0

Криптография

Лекция 4. Цифровые подписи.

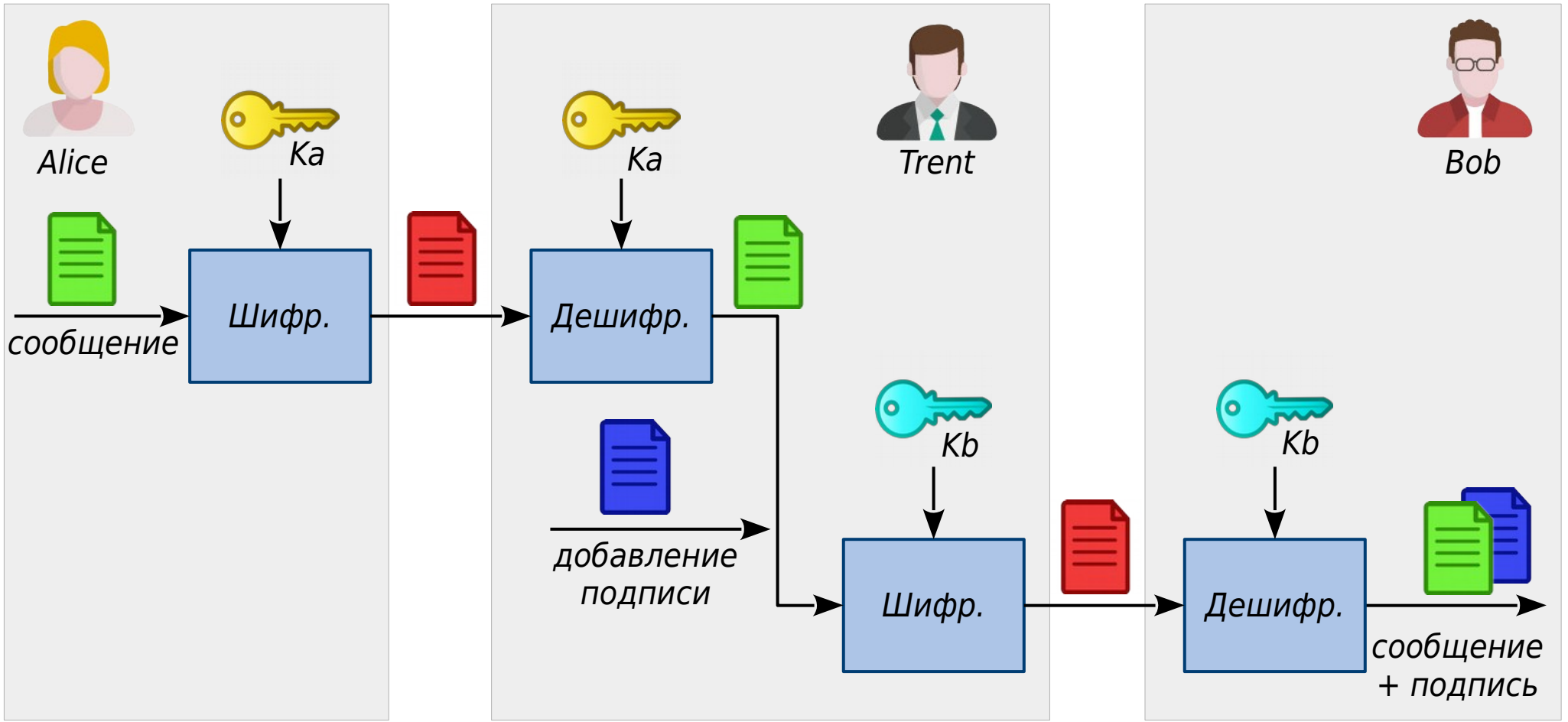
Дмитрий Яхонтов

“Кочерга”, 2019

Требования к цифровой подписи

- Подпись удостоверяет автора сообщения
  именно автор подписи, и никто иной,
  сознательно подписал документ
- Подписанный документ нельзя изменить
  любое изменение документа приводит к тому,
  что подпись становится недействительной
- Подпись нельзя использовать повторно
  она является частью документа, перенести
  подпись на другой документ невозможно
- От подписи невозможно отречься
  автор не может сформировать отказ от своей подписи
  или утверждать, что подпись создана не им.

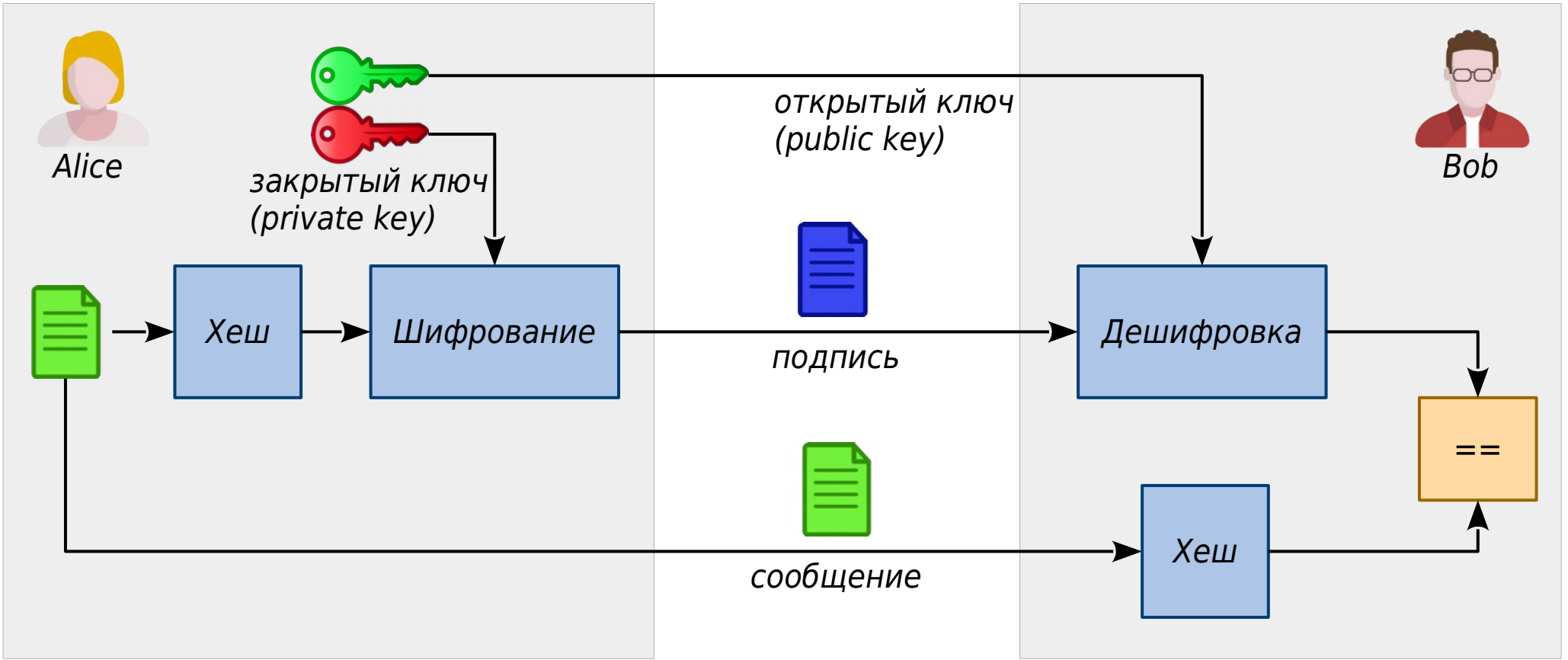
Симметричная схема (с посредником)



Недостатки симметричной схемы

- Нужны защищенные каналы для обмена закрытими ключами между Трентом и каждым из остальных участников
- Если Боб получил от Алисы подписанное сообщение, он может продемонстрировать подпись кому-то еще только через Трента
- Трент должен хранить базу сообщений, либо пересылать Бобу копию шифрованного сообщения Алисы
- Самое главное:
  необходим Трент — сторона, которой все доверяют

Асимметричная схема



Алгоритм DSA
(Digital Signature Algorithm)

Alice

Bob

открытые простые **p, q** ⟶ **p, q**

открытое **g**: $g^q \bmod p = 1$ ⟶ **g**

например, **g** = $h^{(p-1)/q} \bmod p$

закрытый ключ: **X** < q

открытый ключ: **Y** = $g^x \bmod p$ ⟶ **Y**

сообщение ⟶ *хеш* **H**

секретное случайное **K** < q

подпись:

R = $(g^K \bmod p) \bmod q$ ⟶

S = $(K^{-1}(H + X \cdot R)) \bmod q$ ⟶

сообщение ⟶ *хеш* **H**

проверка подписи:

U1 = $(H \cdot S^{-1}) \bmod q$

U2 = $(R \cdot S^{-1}) \bmod q$

V = $((g^{U1} \cdot y^{U2}) \bmod p) \bmod q$

если **V** = **R**, *то подпись верна*

Почему DSA работает?

- $S = K^{-1}(H + X \cdot R) \bmod q$
  $K = S^{-1}(H + X \cdot R) \bmod q$ — *перенесли* K *в левую часть, а* S — *в правую*
  $K = (H \cdot S^{-1} + X \cdot R \cdot S^{-1}) \bmod q$

- *возведём* g *в степень правой и левой части (по модулю* p*)*
  $g^{K} \bmod p = g^{(H \cdot S^{-1} + X \cdot R \cdot S^{-1}) \bmod q} \bmod p$
  $g^{K} \bmod p = g^{H \cdot S^{-1} \bmod q} \cdot g^{X \cdot R \cdot S^{-1} \bmod q} \bmod p$
  это — U1, *а вот это* — U2
  $g^{X} \bmod q = Y$

- $g^{K} \bmod p = g^{U1} \cdot Y^{U2} \bmod p$

- *возьмём левую и правую части по модулю* q
  $g^{K} \bmod p \bmod q = g^{U1} \cdot Y^{U2} \bmod p \bmod q$
  слева — **R**, *а справа —* **V**

Алгоритм DSA
(пример с числами)

Alice

Bob

p = 29, **q** = 7 ⟶ **p, q**

g = $2^{(29-1)/7}$ mod 29 = 16 ⟶ **g**

закрытый ключ: **X** = 6

открытый: **Y** = 16^{6} mod 29 = 20 ⟶ **Y**

хеш сообщения **H** = 5

случайное **K** = 2

$\mathbf{K}^{-1}$ mod 7 = 4

подпись:

R = (16^{2} mod 29) mod 7 = 3 ⟶

S = (4 (5 + 6 * 3)) mod 7 = 1 ⟶

проверка подписи:

S^{-1} mod 7 = 1

U1 = (5 * 1) mod 7 = 5

U2 = (3 * 1) mod 7 = 3

V = ((16^{5} * 20^{3}) mod 29) mod 7

= (8388608000 mod 29) mod 7

= 3 = **R**

название системы	год	вычислительная задача	примечание
RSA (Rivest-Shamir-Adleman)	1977	разложение на простые множители	
Лэмпорта (Lamport)	1979	любая односторонняя функция	одноразовые пары ключей
ESIGN (Efficient digital SIGNature)	1985	разложение на простые множители	быстрее, чем RSA
Эль-Гамаля (Elgamal)	1985	дискретный логарифм	
Шнорра (Schnorr)	1989	дискретный логарифм	модификация схемы Эль-Гамаля
DSA (Digital Signature Algorithm)	1991	дискретный логарифм	
ECDSA (Elliptic Curve Digital Signature Algorithm)	1999	дискретный логарифм на эллиптич. кривых	
ГОСТ Р 34.10-2012	2012	дискретный логарифм на эллиптич. кривых	

Подсознательный канал (Subliminal Channel)

на примере DSA

- Алиса и Боб выбирают
  Z — закрытый ключ для подсознательного канала
- Алиса подписывает сообщение. Она выбирает случайное число **K** так, чтобы:
  - для передачи 1:
    параметр подписи **R** был квадратичным вычетом по модулю **Z**
    (существует **n** такое, что $R = n^2 \bmod Z$)
  - для передачи 0:
    R не был квадратичным вычетом по модулю **Z**
- Боб проверяет подпись, а затем восстанавливает переданный бит из параметра **R**, зная **Z**
- Передать несколько бит можно, используя сразу несколько модулей **Z**

Уничтожение подсознательного канала

- Число **K** должно генерироваться совместно обеими сторонами
- Алиса не должна контролировать ни один бит числа K
- Боб не должен узнать ни один бит числа K
- Боб должен иметь возможность проверить, что для подписи использовалось именно сгенерированное K

Alice

выбирает **k1**

u = g^{k1} mod p ⟶ **u**

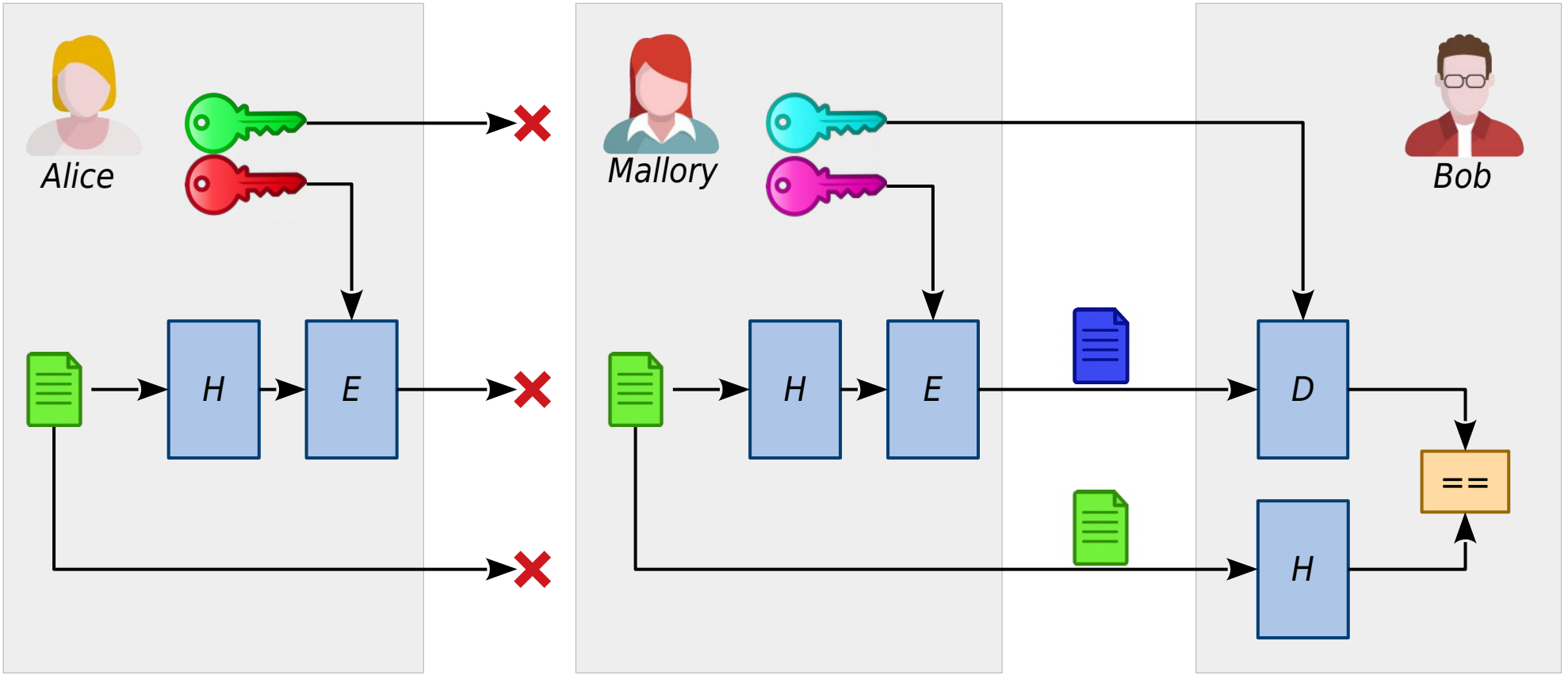
k2 ⟵ *выбирает* **k2**

K = k1*k2 mod (p-1)

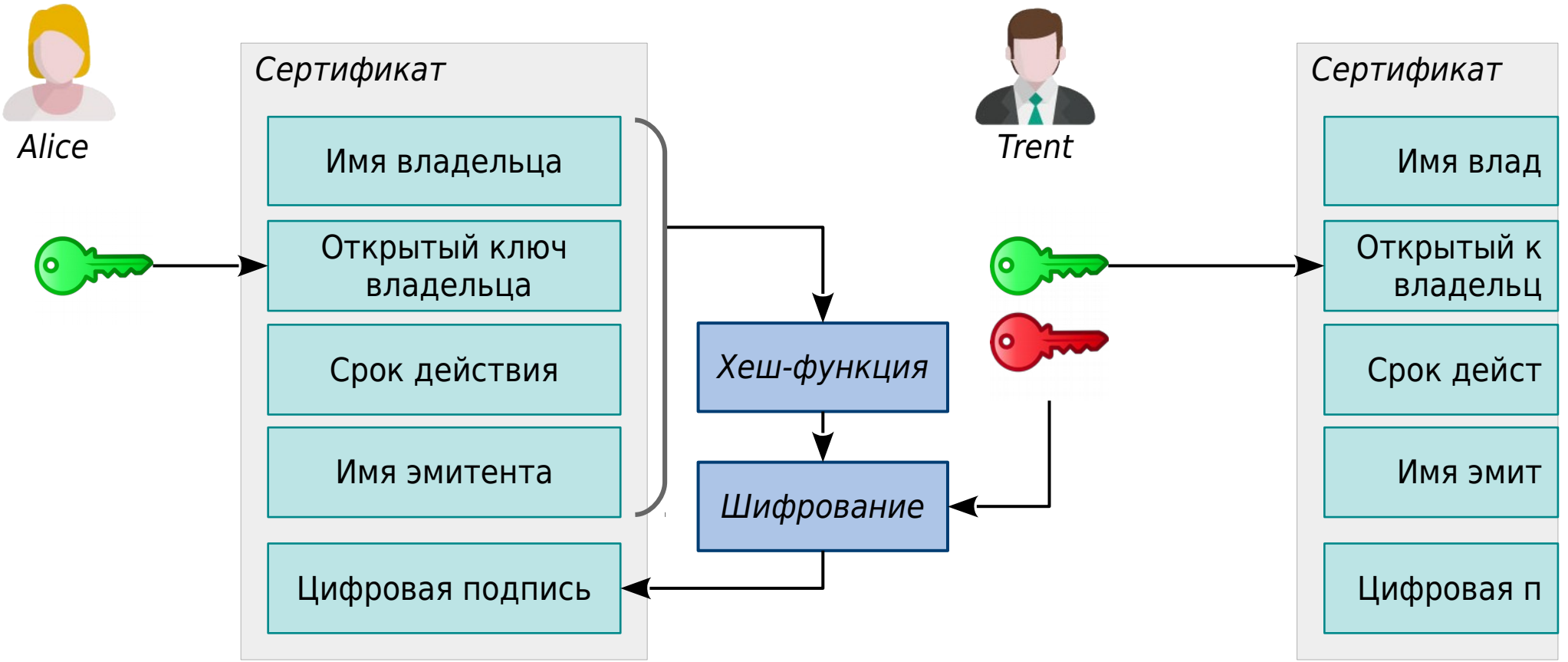
проверяет, что
(**u**^{k2} mod **p**) mod **q** = **R**

Bob

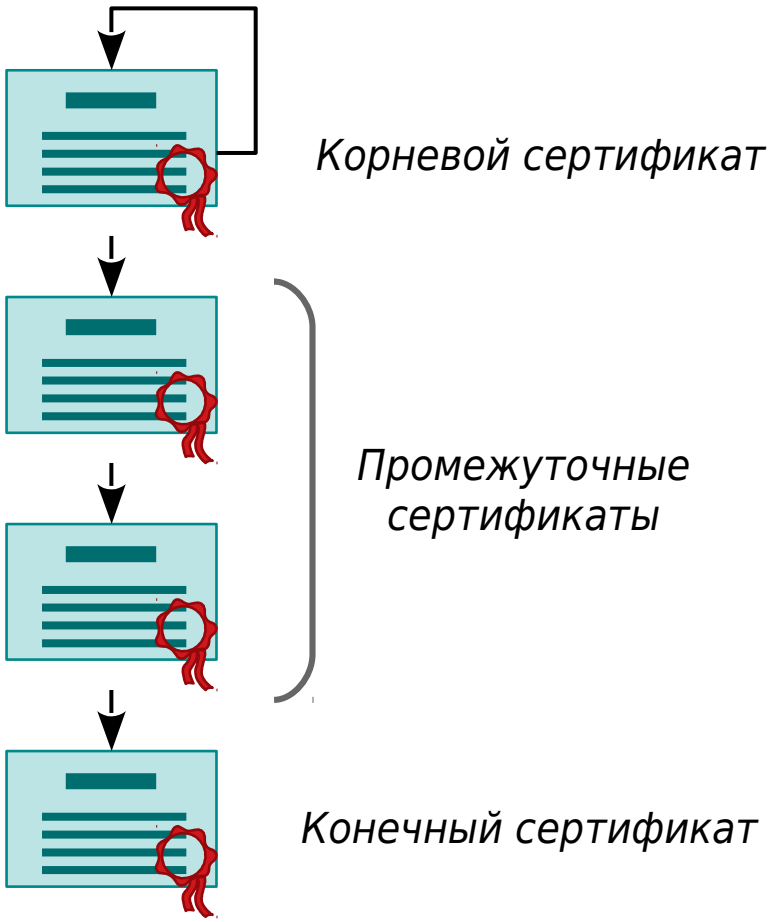
Атака “человек посередине”
(Man-in-the-Middle, MitM)



Сертификат открытого ключа



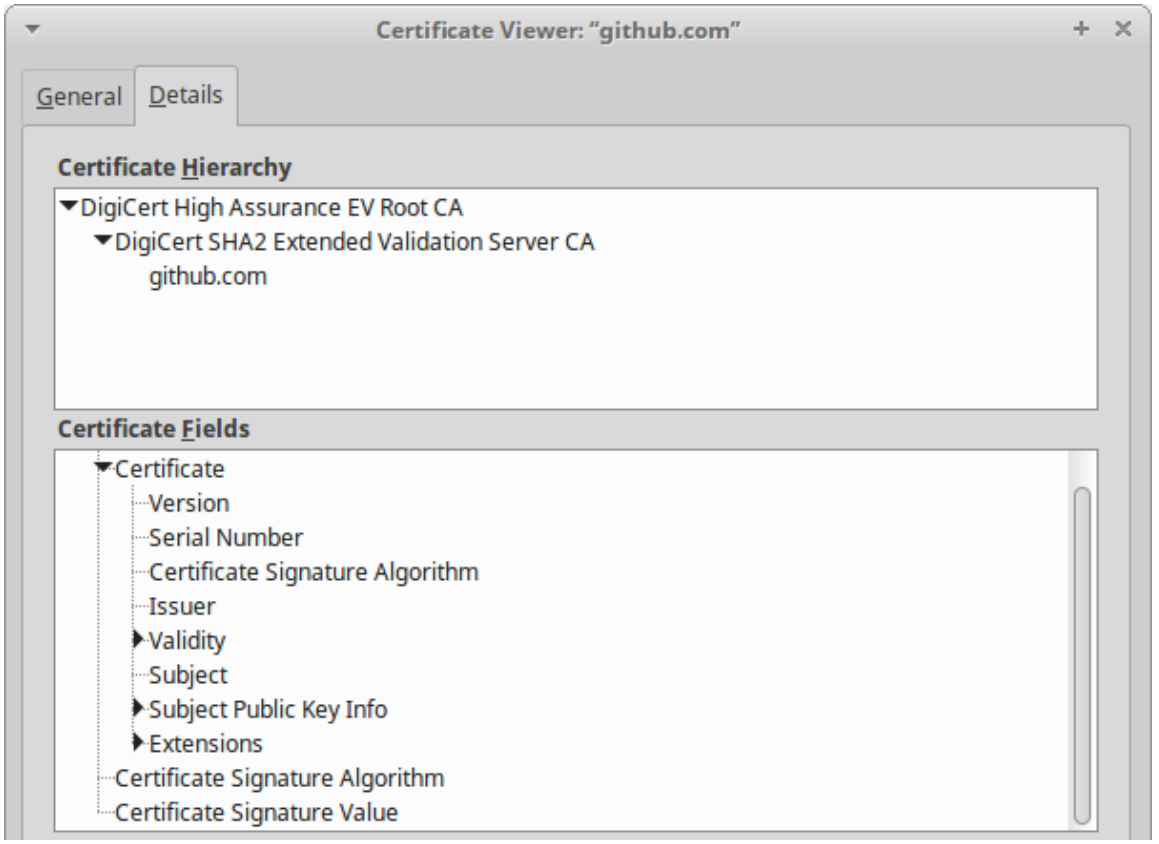
Цепочка доверия



Корневой сертификат

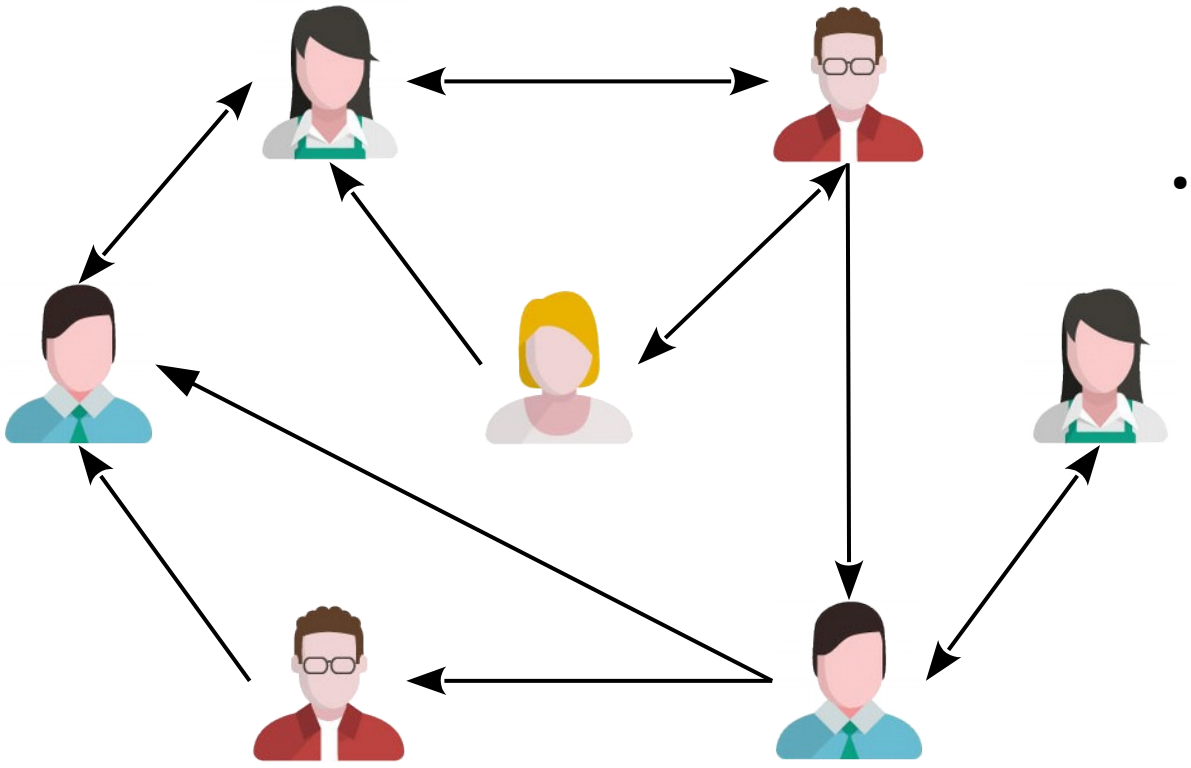
Промежуточные сертификаты

Конечный сертификат



Пример: цепочка сертификатов в протоколе TLS

Сеть доверия
(Web of Trust, WoT)



- Принадлежность открытого ключа участнику сети удостоверяют другие участники
- Возможны различные уровни доверия:
  1. *“Я его знаю”*
  2. *“Я знаю того, кто его знает”*
  3. *“Я знаю того, кто знает того, кто его знает”* и так далее.

WOT® WEB OF TRUST — *это не оно :)*

Задача*

Трент поручил Бобу разработать интерактивный телефонный справочник. Боб написал программу, которая на запрос *{имя}* отвечает парой *{имя, номер_телефона}*, сопровождаемой *подписью* Трента. Подпись доказывает, что абонент с определенным именем действительно имеет определенный номер телефона.

Помогите Бобу модернизировать программу так, чтобы она дополнительно возвращала доказательство отсутствия в справочнике абонента с определенным именем. Доказательство должно представлять собой структуру данных, подписанную Трентом.

Трент опасается передавать Бобу личный ключ подписи, поэтому все доказательства должны быть созданы Трентом заранее, в момент формирования справочника.

Ссылки

- Обратная связь:
  - android.ruberoid@gmail.com
  - @androidruberoid
- Анонсы:
  - facebook.com/kocherga.club
  - vk.com/kocherga_club
  - vk.com/kocherga_prog
- Материалы лекций:
  - github.com/notOcelot/Kocherga_crypto
- Видео:
  - youtube.com/channel/UCeLSDFOndl4eKFutg3oowHg