



CFT
experience

Прикладная криптография для земель

Владимир Плизга
Дирекция prepaid карт

```

0000: 10 00 01 02 01 00 90 C2 ED CB 3D 5D 23 5C 59 C7 .....=]#\V\
0010: 03 BF 82 59 CE E1 32 1E 7B 6B EF 78 92 74 8F 7F ...Y...2...k...k...
0020: 20 31 36 52 6C 37 3A 17 02 67 EC 41 C5 20 3F 0D .6R).....g...A...P\
0030: 06 92 76 50 22 D8 86 7C 85 FD C8 1E 48 6E 0B BA ...D.....M...
0040: AD F4 CD 39 33 EF 8C 5B 1A 1A 60 E6 0C 28 C7 7D .3...[... (..
0050: AF B5 09 27 0C 69 F9 1A 8F 78 59 EC 9C AD 9D ED ...1.....Y
0060: 8D 95 35 32 71 5B 6F 6A DC 95 JA VA 2E D1 96 F1 ..5.g... ..
0070: 37 8F 06 1A B2 DF DA 9A 3C 40 30 A1 DB 5C BC 1B 7.....<00... \..
0080: 22 0A DD AC 2D C7 F5 3C 88 62 5C AD 45 61 49 0E ".....<.b\...aI..
0090: 75 69 BF 5D 4C 6D D2 94 3E 26 BC 5F 71 56 7C F1 u1...m...>...V...
00A0: 4A 7R 37 35 30 07 02 EC F2 E6 39 6C 12 08 2C 35 3.7.8.....91...5
00B0: 44 7R 43 D4 C9 25 28 58 F8 26 B8 94 BD DA CD 12 D.C...%(\...
00C0: 09 20 47 BR 4F F8 4B 1B A8 B3 8C D1 28 81 78 7B .G.O...k...k...

```

Обо мне

Инженер-программист (Java)

Дирекция prepaid карт (с 2011 г.)

Неофициальный консультант по SSL в Отделе

Автор:

- утилиты Certificate Watcher (ЦФТ)
- статьи <https://focus.cft.ru/x/e4N8>
- статьи <https://habr.com/post/254205/>
- статьи <https://xakep.ru/2015/08/14/log-almighty/>
- утилиты <https://github.com/Toparvion/nss-java-maker>



Вместо плана

1

- Схемы шифрования

2

- Распространение публичных ключей

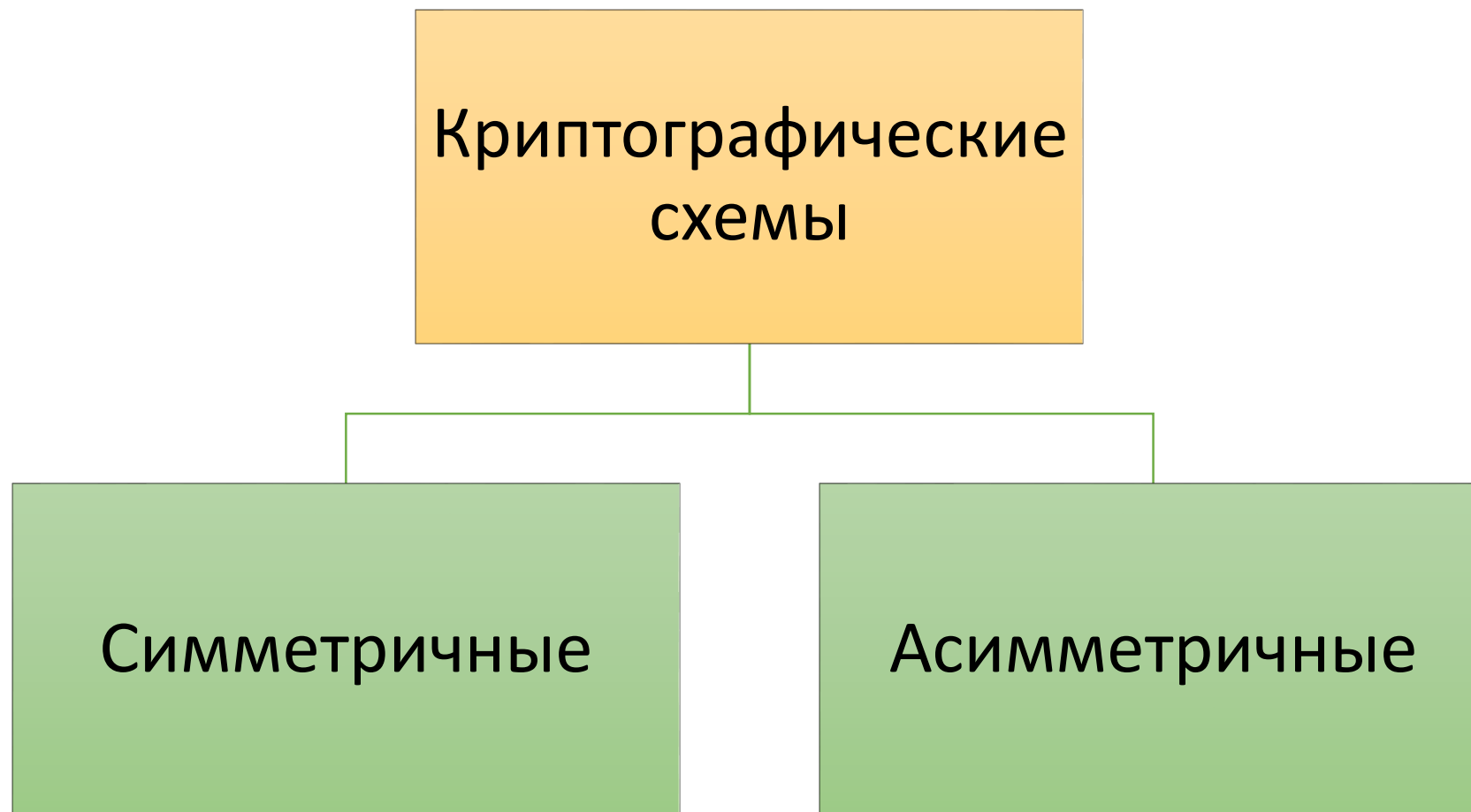
3

- Протоколы SSL/TLS

Схемы шифрования

Симметричные и не очень

Сенсация



Решаемая задача

Передать сообщение от А к В так,
чтобы **в случае перехвата** было
максимально сложно его прочитать

Симметричная схема

AES, Blowfish, 3DES,
Serpent, RC4, ChaCha

← Интернет →

code=hJ2&s2



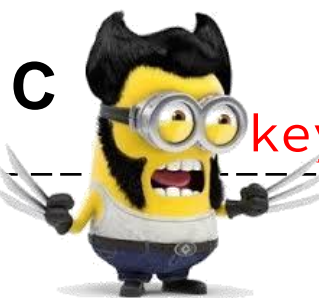
A

code = encrypt(message, key)



B

message = decrypt(code, key)



C

key=password

Можно передавать по сети
Нельзя передавать по сети

Асимметричная схема

Основа – не 1, а 2 ключа:

Однозначно связаны между собой

Но нельзя сложно вывести один из другого

Где взять такую модель?

Основная теорема арифметики:

Любое натуральное число
однозначно представимо
произведением простых чисел



Асимметричная схема

$$21 = \boxed{?} * 3$$

ВЫВЕСТИ

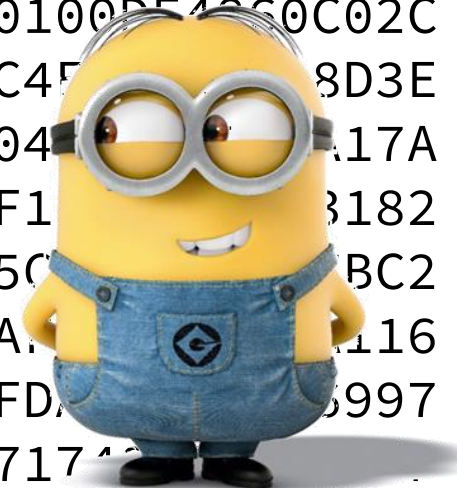
Верно!
Почти...

Так ведь
можно
подобрать!

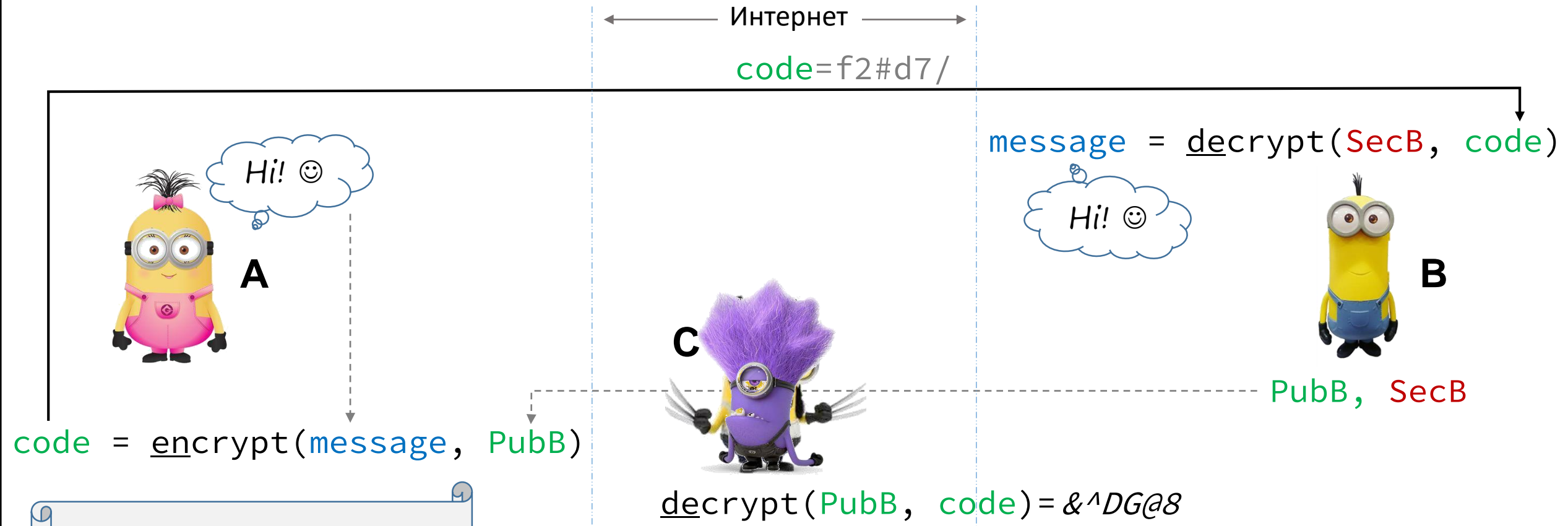


Пример настоящего ключа

0x30820222300D06092A864886F70D01010105000382020F003082020A0282020100D54260C02C
4C112206873D606B8F9B53CAF56AEE35D962DD2EF8539730CF88201EAED45ECEC4F8D3E
9DC2702B4EBA3E002C604FEFE29C02139C21B5073ECE9CB6FBF05226641189FD04A17A
A4CD8066E3C41FFBA707F3FD3EBC75762474796FD2EBFC80A9BA65EA311DADCFF1B182
36CEB5B8D81A19393547274E5A995B1AC92765BD0B115959EAE05B39993A11150BC2
704653F2159DA85C0A7A55A391BD82C58EB826B882A834BB5C6E4B78668A19B5A1116
488301E7FDE39A2D7681A001F7172D55BA35D7C143C33D112D213BCA144B19E3FD1997
4C354F4E4D7B4EBF9A5FA682934B383EE093EE43491EAA52A4386B0DB96D9EF471712
DB8DCC42A8A0A8905838FFED4DE0251D50CE31F819EB7E8445E53521B0FDA72BEF7DF5C11C4370
BB60CC3029CFAB6AF59255C0D10AF9AAC18D828B2E7B0C7E32248074D90ED0D08A901F937E6054
07751875CE05BF3D0B30F5BB8849B3F050A4C7C604D5DA4E4206139A68CD7E87B52E7F66DC930F
3C8824700A934A5A385B03117C0FCDBFC05B14919473100AE8165C6F80AAABCC7182A0881AEBE6
D9B99C696AD0F618683969543AACABFD7DB8F4AC6CC6254366300452C91BEB9B147D7EAF1167D8
2CA7FE2176048F8A10D605C0809F015B6A49858116F4F33D2AC0972BE0FFEF6D62A505377E3647
0203010001



Асимметричная схема: шифрование



Можно передавать по сети
Нельзя передавать по сети

Шифрование: попутное резюме

1. Используются ключи **только получателя**
2. **Публичный** ключ используется **для шифрования**,
а **приватный** – **для дешифрации**
3. Для обратной передачи схема **зеркальна**



Проверка боем

**SCIENTIFIC
AMERICAN**



Колонка «Математические игры»

Задача: расшифровать сообщение,
зашифрованное алгоритмом RSA (425 бит)

Rivest, Shamir, Adleman



Ronald Rivest

Проверка боем

Шифр

9686	9613	7546	2206
1477	1409	2225	4355
8829	0575	9991	1245
7431	9874	6951	2093
0816	2982	2514	5708
3569	3147	6622	8839
8962	8013	3919	9055
1829	9451	5781	5154

Публичный ключ

n=114381625757888867669
23577997614661201021829
67212423625625618429357
06935245733897830597123
56395870505898907514759
9290026879543541
e=9007

Призовой фонд \$100

Проверка боем



Ronald Rivest

- 1977 год
- 40 квадриллионов лет на факторизацию числа n

VS



Arjen Lenstra

- 1993 год
- 6 месяцев
- 600 добровольцев
- 1600 машин
- «Квадратичное решето»

Проверка боем



Ronald Rivest



Arjen Lenstra

VS

➤ 1977 год

➤ на фактор

THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE

Волшебные слова — брезгливый ягнятник

➤ «Квадратное решето»

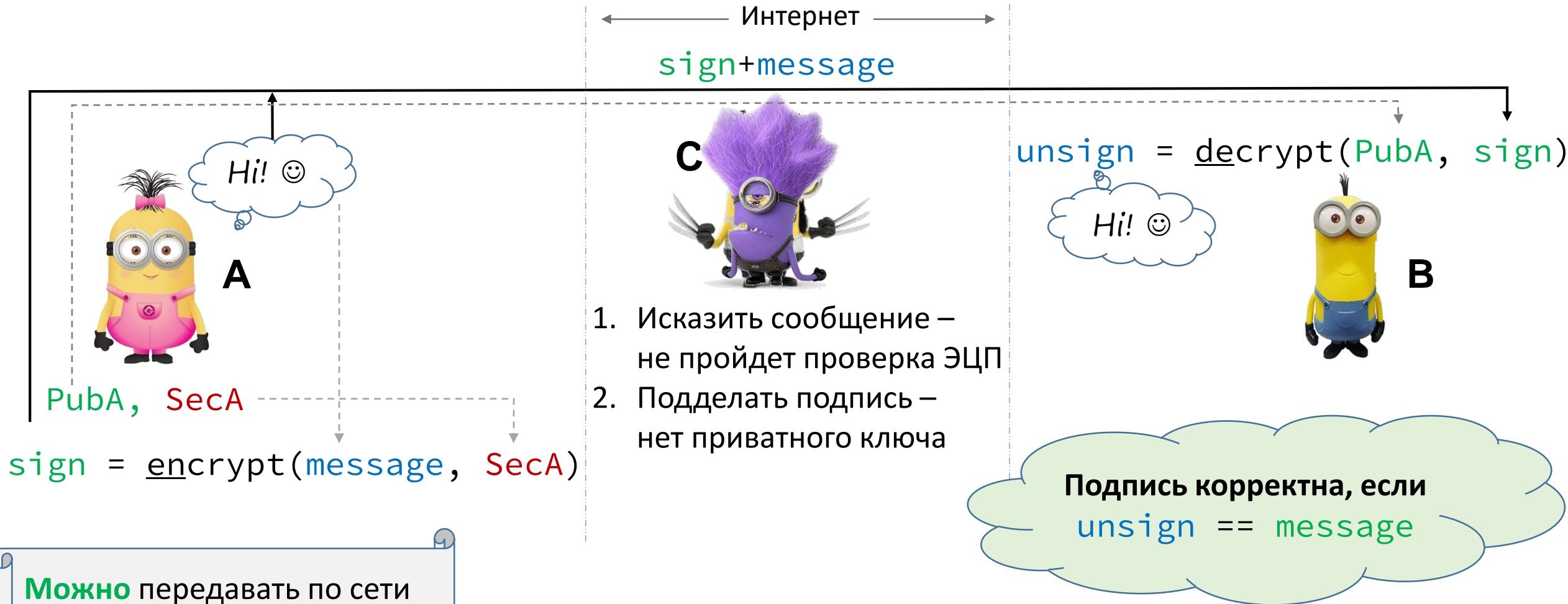
Электронная цифровая подпись (ЭЦП)

Решаемая задача

Убедить получателя в том, что
полученное им сообщение:

1. Не искажено при передаче;
2. Создано именно отправителем.

Асимметричная схема: ПОДПИСЬ



Можно передавать по сети
Нельзя передавать по сети

Подпись: попутное резюме (1/2)

1. Используются ключи **только отправителя**
2. **Приватный** ключ используется **для шифрования** («подписывания»),
а публичный – **для дешифрации** (проверки)
3. Для обратной передачи схема **зеркальна**

По сравнению
с асимметричным шифрованием
здесь всё наоборот



Подпись: попутное резюме (2/2)

В основе – факт **секретности** приватного **ключа** у **отправителя**

Функция схемы – **только индикаторная**, т.е. не позволяет:

- Определить характер сбоя (умышленный/случайный)
- Восстановить искаженное сообщение



Поправка на реальность: хэширование

Подпись рассчитывается от **дайджеста** сообщения.

Дайджест = **хэш** – строка **фиксированной** длины, **однозначно** описывающая исходное сообщение.

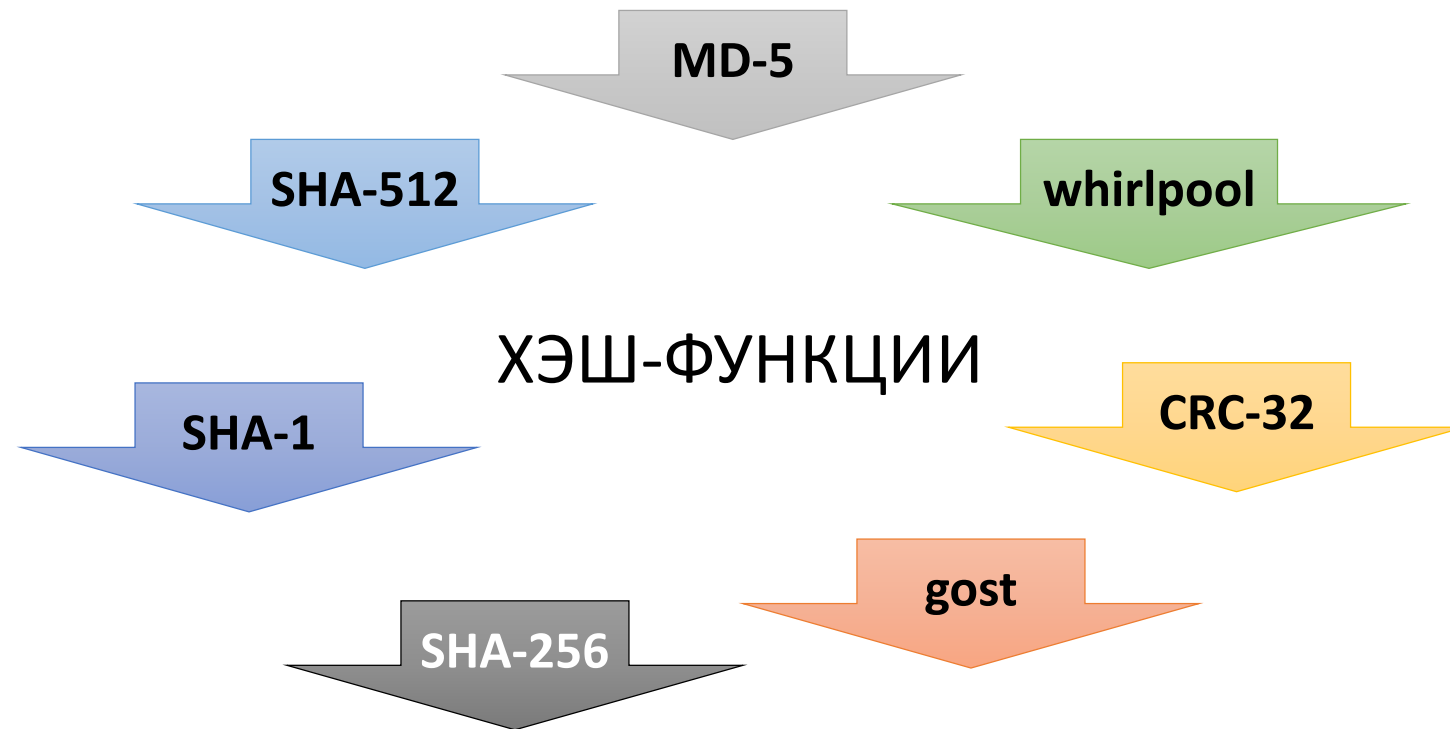
Например:



```
9b99fe3b2c6837f97778dc4d027a49a4  
345f32e654d9b84141cfe7d1a46401ae
```



Поправка на реальность: хэширование



Хэширование

Хэширование – необратимое преобразование сообщения в число фиксированной разрядности, сильно зависящее от исходного сообщения.

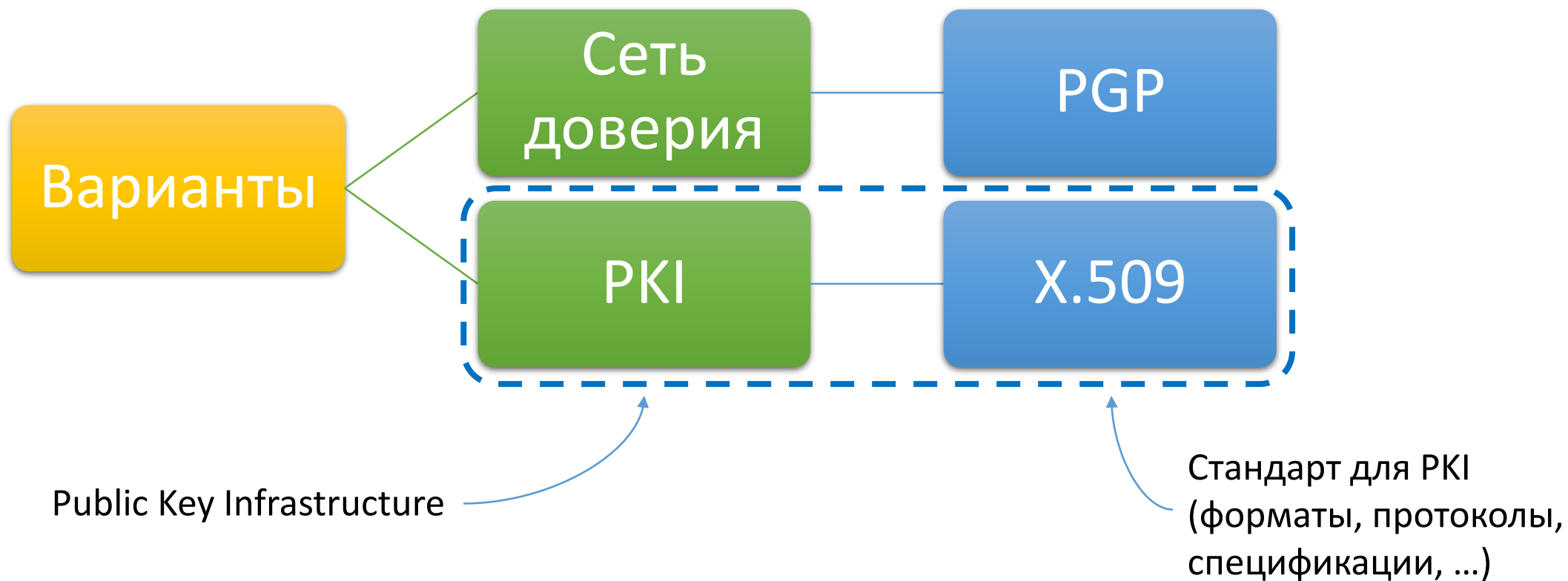
Хэширование в ЭЦП делает размер подписи независимым от размера исходного сообщения.



Распространение ключей

Публичные ключи на публике

Как надежно распространять ключи?



Что для этого нужно?

1. Доверенная третья сторона
2. Механизм для «закрепления» доверия

Удостоверяющий
центр (УЦ)



Цифровая подпись



Удостоверяющий центр (УЦ)

★ **Удостоверяющий центр (УЦ), центр сертификации** - сторона (отдел, организация), чья честность неоспорима, а **открытый** ключ широко известен. (© Wikipedia)

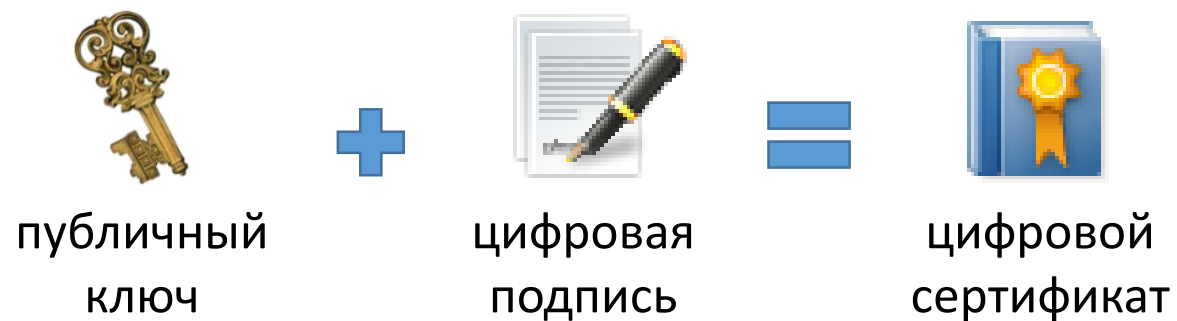
Примеры: Google, Thawte и любой желающий



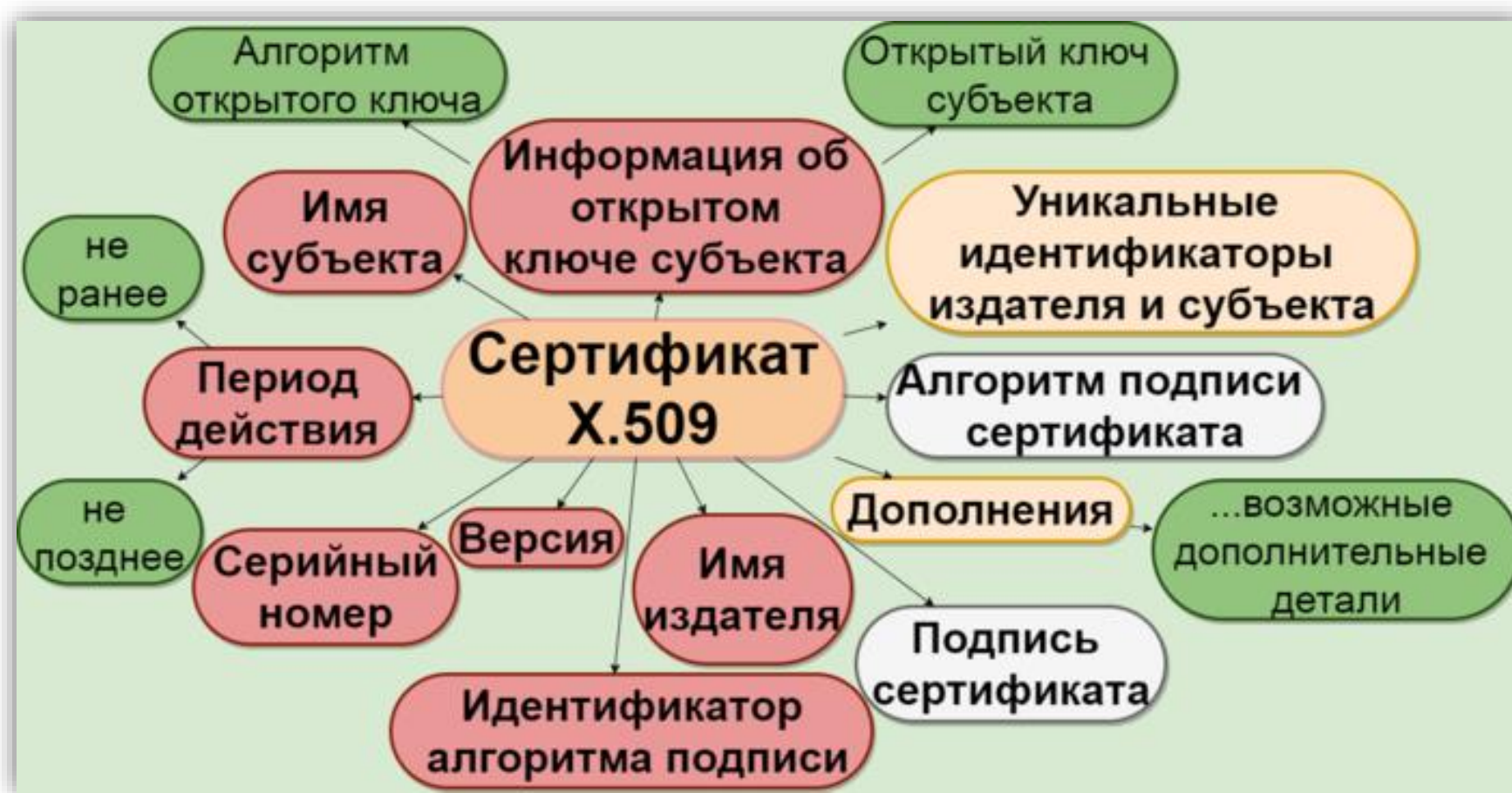
Механизм «закрепления» доверия

УЦ заверяет **чужой публичный** ключ путем его «подписывания» **своим закрытым** ключом.

Формат связки ключа и его подписи:



А как на самом деле



https://commons.wikimedia.org/wiki/File:Структура_сертификата_X.509.png

Защита от компрометации сертификата

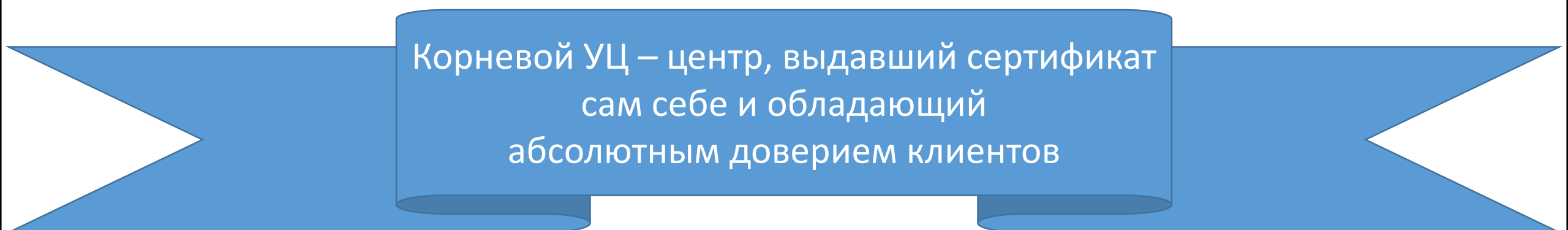
Сертификат содержит **публичный ключ** УЦ

=> ключ может быть **скомпрометирован**

=> ключ нуждается **в защите**

=> ключ нужно **подписать** в другом УЦ

=> а его ключ – в **другом УЦ**, и так далее до...



Корневой УЦ – центр, выдавший сертификат сам себе и обладающий абсолютным доверием клиентов

Проверка сертификатов клиентом

Клиенты не обязаны знать сертификаты всех серверов

Если клиент не доверяет серверу,
то он может проверить **его УЦ**

А если не может доверять его УЦ,
то может проверить **УЦ его УЦ**

А если не может доверять УЦ его УЦ,
то может проверить **УЦ его УЦ его УЦ...**

И так пока не найдёт доверенный **корневой УЦ**

обычно ≈3 шагов



Хранилища корневых сертификатов

Chrome, IE, Opera

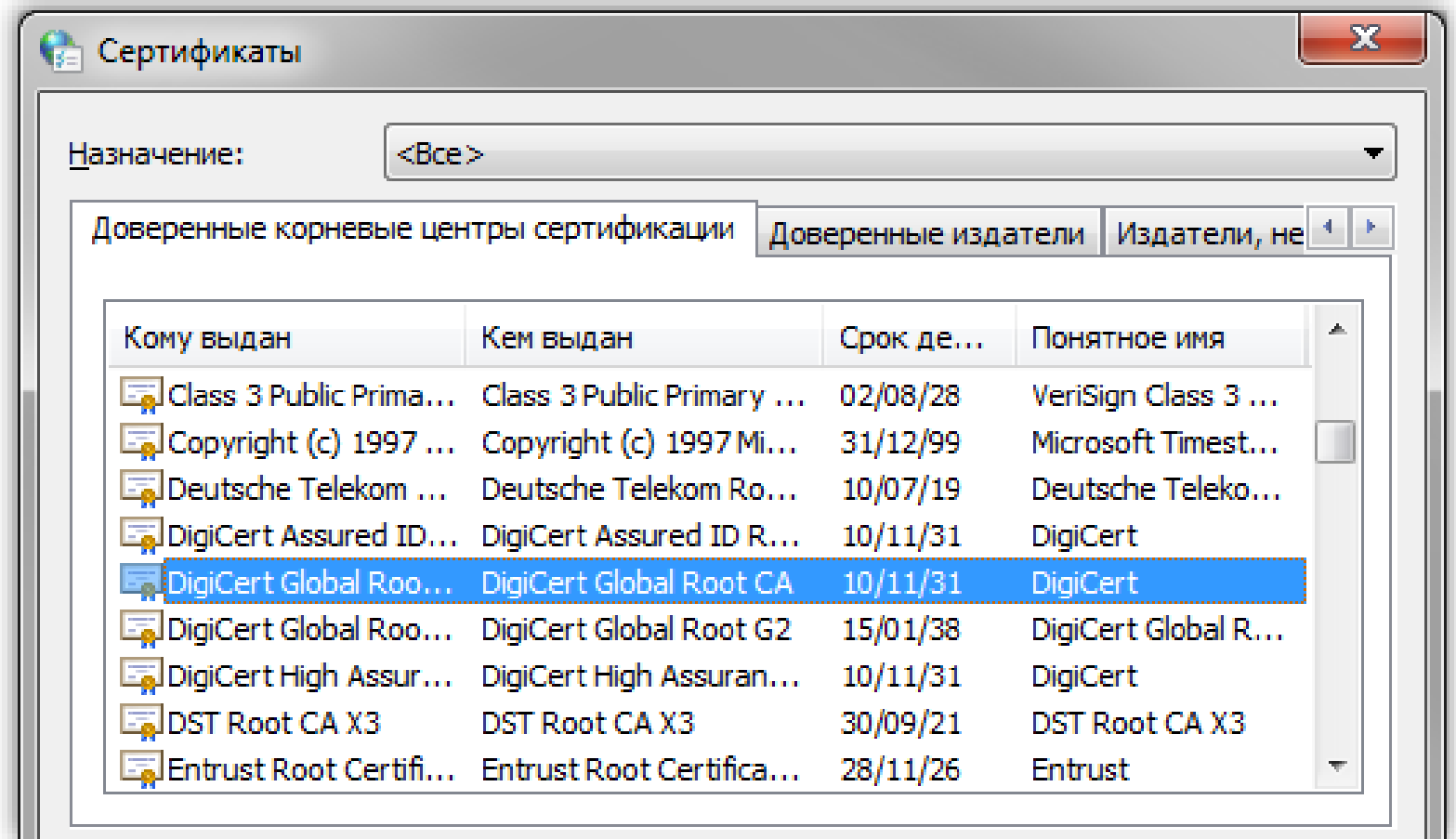
В Windows:

Панель управления →

Свойства браузера →

Содержание →

Сертификаты



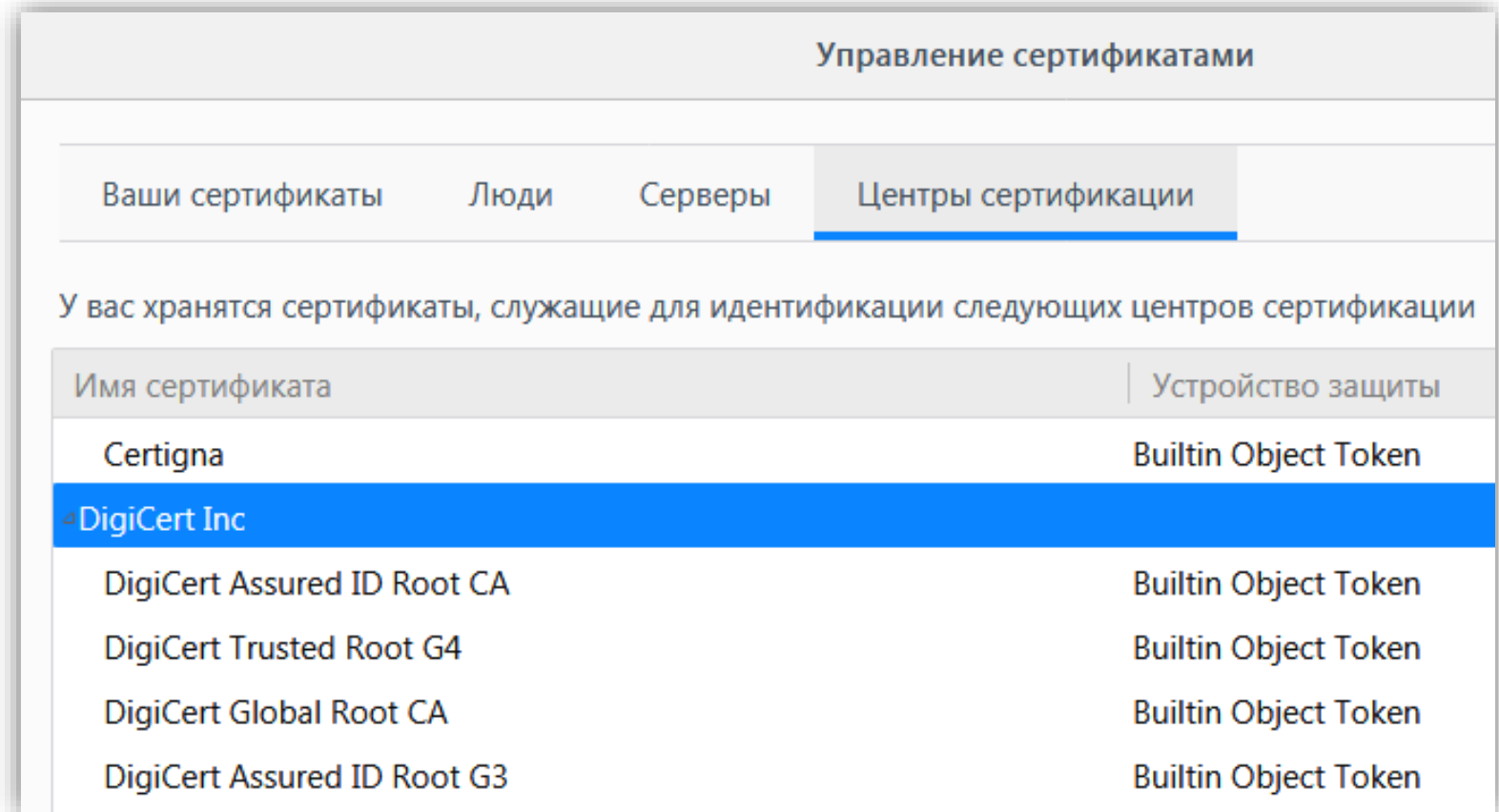
Хранилища корневых сертификатов

Firefox

Настройки →

Приватность и защита →

Просмотр сертификатов



Управление сертификатами

Ваши сертификаты Люди Серверы **Центры сертификации**

У вас хранятся сертификаты, служащие для идентификации следующих центров сертификации

Имя сертификата	Устройство защиты
Certigna	Builtin Object Token
▾ DigiCert Inc	
DigiCert Assured ID Root CA	Builtin Object Token
DigiCert Trusted Root G4	Builtin Object Token
DigiCert Global Root CA	Builtin Object Token
DigiCert Assured ID Root G3	Builtin Object Token

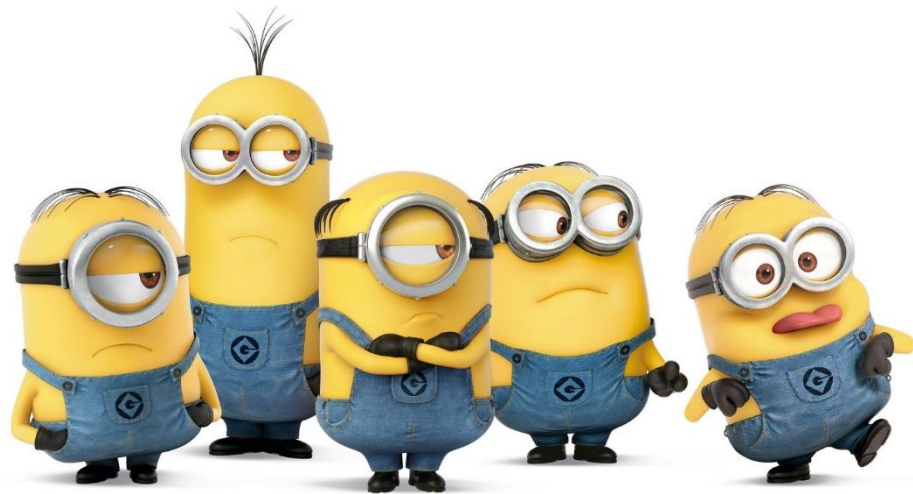
Что мы имеем?

- ✓ Шифрование данных при передаче (RSA, EC)
- ✓ Защита ключей от подделки (ЭЦП)
- ✓ Механизм распространения ключей (PKI + сертификаты X.509)
- Применимость в масштабах Интернета
(где с **миллионами** серверов общаются **миллиарды** клиентов)
- Производительность





И как быть?

1. Шифровать прикладные сообщения **симметричным** алгоритмом
2. Генерировать **новый** ключ для каждой **новой** сессии



Установка защищенного канала

1. Клиент обращается к серверу
2. Сервер выдает клиенту свой **публичный ключ** в составе сертификата
3. Клиент проверяет ЭЦП сертификата
4. Клиент генерирует случайный симметричный **сессионный ключ**
5. Клиент шифрует его **публичным ключом** сервера и **отправляет** серверу 
6. Сервер вскрывает сессионный ключ своим **приватным ключом** 
7. Клиент шифрует все сообщения серверу **сессионным ключом**
8. Сервер шифрует все сообщения клиенту **сессионным ключом**

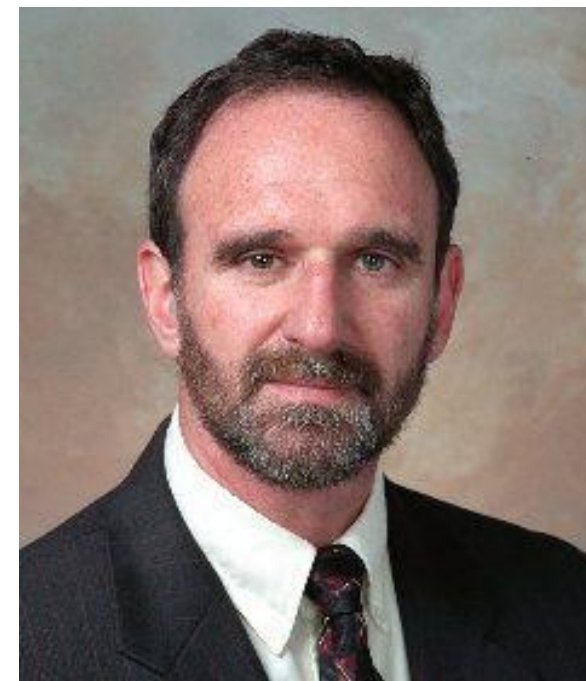
Алгоритм Диффи-Хеллмана (DH)



У. Диффи

Позволяет двум сторонам выработать общий ключ, никогда не передавая его друг другу

=> Приватный ключ сервера перестаёт быть «Ахиллесовой пятой»



М. Хеллман

SSL/TLS

И другие ругательства

Протокол SSL/TLS

★ Описанная схема – основа SSL/TLS

Соединение по SSL/TLS начинается с рукопожатия (handshake):

Обмен сертификатами, их проверка

Выбор шифронабора (AES, EC, ...)

Выработка сессионного ключа

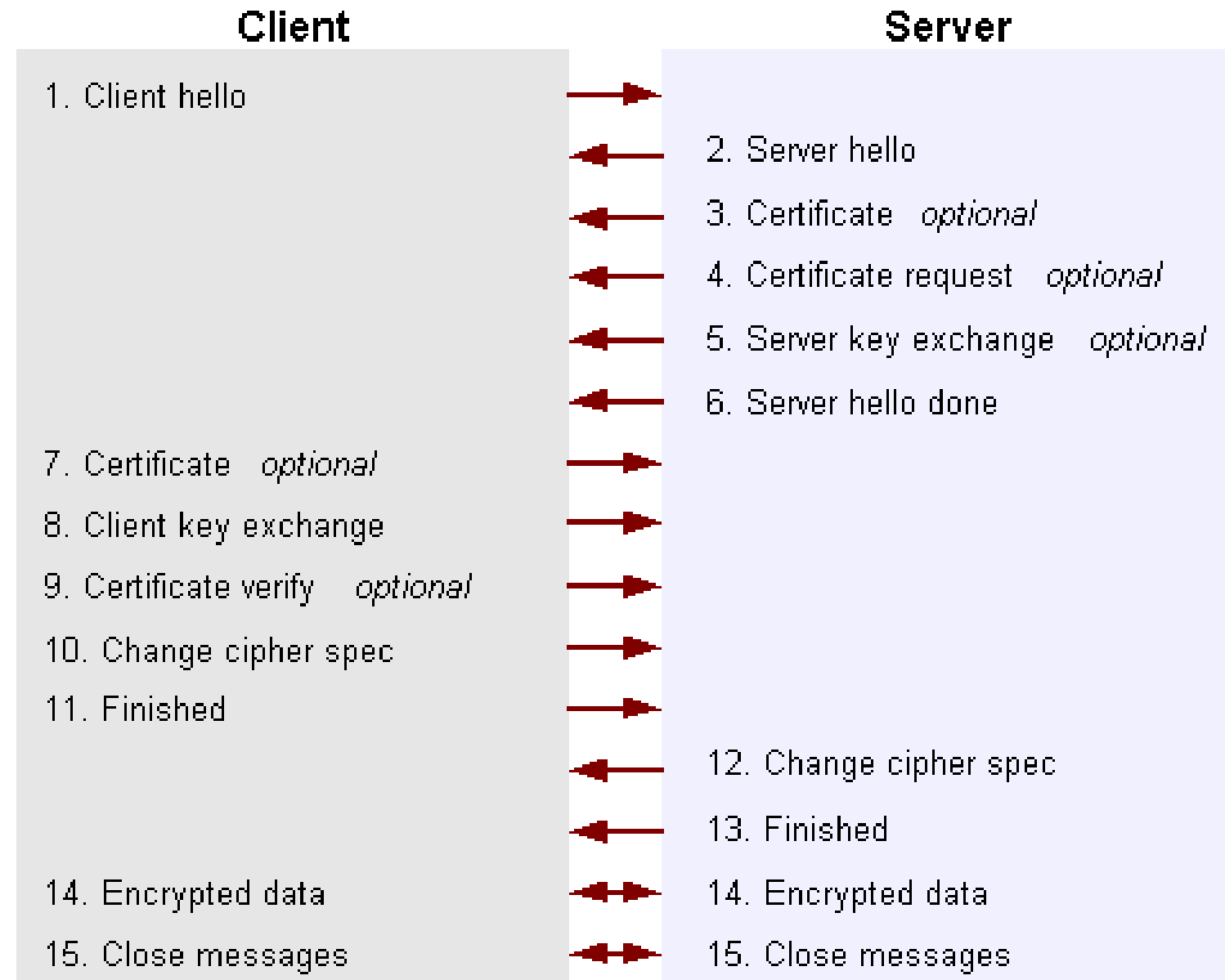
Тестирование канала



Обмен сообщениями в SSL



SSL Messages



<https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html#HowSSLWorks>

Протоколы SSL и TLS: в чем разница?

SSL ([англ. Secure Sockets Layer](#) — уровень защищённых [сокетов](#)) — [криптографический протокол](#), который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений. Протокол широко использовался для обмена мгновенными сообщениями и передачи голоса через [IP](#) ([англ. Voice over IP](#) — [VoIP](#)) в таких приложениях, как [электронная почта](#), интернет-факс и др. В 2014 году правительство США сообщило об уязвимости в текущей версии протокола^[1]. SSL должен быть исключён из работы в пользу [TLS](#) (см. CVE-2014-3566).

TLS ([англ. transport layer security](#) — Протокол защиты транспортного уровня^[1]), как и его предшественник [SSL](#) ([англ. secure sockets layer](#) — слой защищённых сокетов), — [криптографические протоколы](#), обеспечивающие защищённую передачу данных между узлами в сети [Интернет](#)^[2]. TLS и SSL используют [асимметричное шифрование](#) для аутентификации, [симметричное шифрование](#) для конфиденциальности и [коды аутентичности сообщений](#) для сохранения целостности сообщений. TLS-протокол основан на спецификации протокола [SSL](#) версии 3.0, разработанной компанией [Netscape Communications](#)^[3]. Сейчас развитием стандарта TLS занимается [IETF](#). Обновления протокола были в [RFC 5246](#) (август 2008), [RFC 6176](#) (март 2011) и [RFC 8446](#) (август 2018).

Короче, это одно и то же.

Резюме





CFT
experience

Прикладная криптография для земель

Владимир Плизга

 @toparvion

 toparvion@gmx.com