

Современные криптографические протоколы

Протоколы SSL/TLS

Что такое SSL / TLS?

- Протокол Transport Layer Security
 - Основан на Secure Sockets Layers, версия 3.0
 - Похожий дизайн, разные алгоритмы
 - Стандарт де-факто для криптографической защиты веба
 - “Основная задача TLS обеспечить конфиденциальность и целостность коммуникаций между двумя приложениями”
 - ... против «человека в середине»
 - Используется для защиты веб-серверов, VoIP, email, и т.д.
 - Есть практически в любом браузере
-

SSL / TLS в реальном мире

Wells Fargo Account Summary - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Favorites Print Home

Address https://online.wellsfargo.com/mn1_aa1_on/cgi-bin/session.cgi?sessargs=coAn76ax52xlpX8uoCT8rRBFMMdJldx Go Links Yahoo maps Mapblast Dictionary

Home | Help Center | Contact Us | Locations | Site Map | Apply | Sign Off

WELLS FARGO

Account Summary Last Log On: January 06, 2004

> Account Summary

- Brokerage
- Bill Pay
- Transfer
- Account Services
- My Message Center

Stay organized with FREE 24/7 access to Online Statements. Sign up today.

Sign up for the Wells Fargo Rewards® program and get 2,500 points. Learn More.

Wells Fargo Accounts **OneLook Accounts**

Tip: Select an account's balance to access the Account History.

NEW [Enroll for Online Statements](#) [My Message Center](#)

Cash Accounts

Account	Account Number	Available Balance
Checking Add Bill Pay		
Total		

To end your session, be sure to Sign Off.

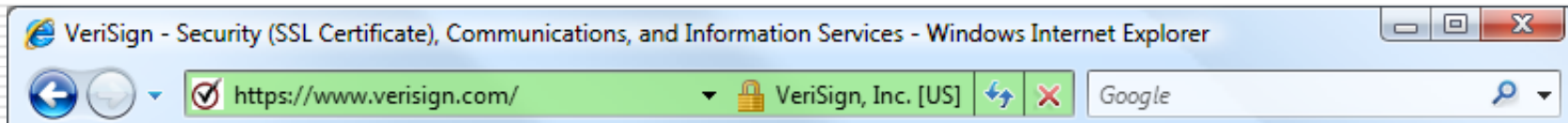
Account Summary | Brokerage | Bill Pay | Transfer | My Message Center | Sign Off
Home | Help Center | Contact Us | Locations | Site Map | Apply

© 1995 - 2003 Wells Fargo. All rights reserved.

Internet



Что означает этот «замочек»?



- Что веб-страница загружена с того узла, который указан в адресной строке
 - Но пользователь должен уметь интерпретировать ту информацию, которую он получает от браузера
 - Содержимое страницы не простматривалось и не модифицировалось «человеком в середине»
-

Назначение SSL / TLS

- SSL/TLS обеспечить безопасность коммуникаций в условиях присутствия злоумышленника в сети
 - Вы сидите с ноутбуком в интернет-кафе, подключенном к контролируемому спецслужбами провайдеру во враждебной авторитарной стране (попытайтесь этого избегать ;-)
 - Злоумышленник владеет сетью
 - WiFi, DNS, маршрутизаторы, его собственные веб-сайты
 - Может прочитать или изменить любой пакет, внедрить собственные пакеты в сессию
-

История SSL

- SSL 1.0
 - Внутренняя разработка Netscape, начало 1994?
 - Не использовался в реальности
 - SSL 2.0
 - Опубликован Netscape в ноябре 1994
 - Имеет несколько уязвимостей
 - SSL 3.0
 - Разработка Netscape и Paul Kocher, ноябрь 1996
 - TLS 1.0
 - Общий стандарт, основанный на SSL 3.0, январь 1999
 - Несовместим с SSL 3.0
 - TLS использует HMAC вместо MAC
-

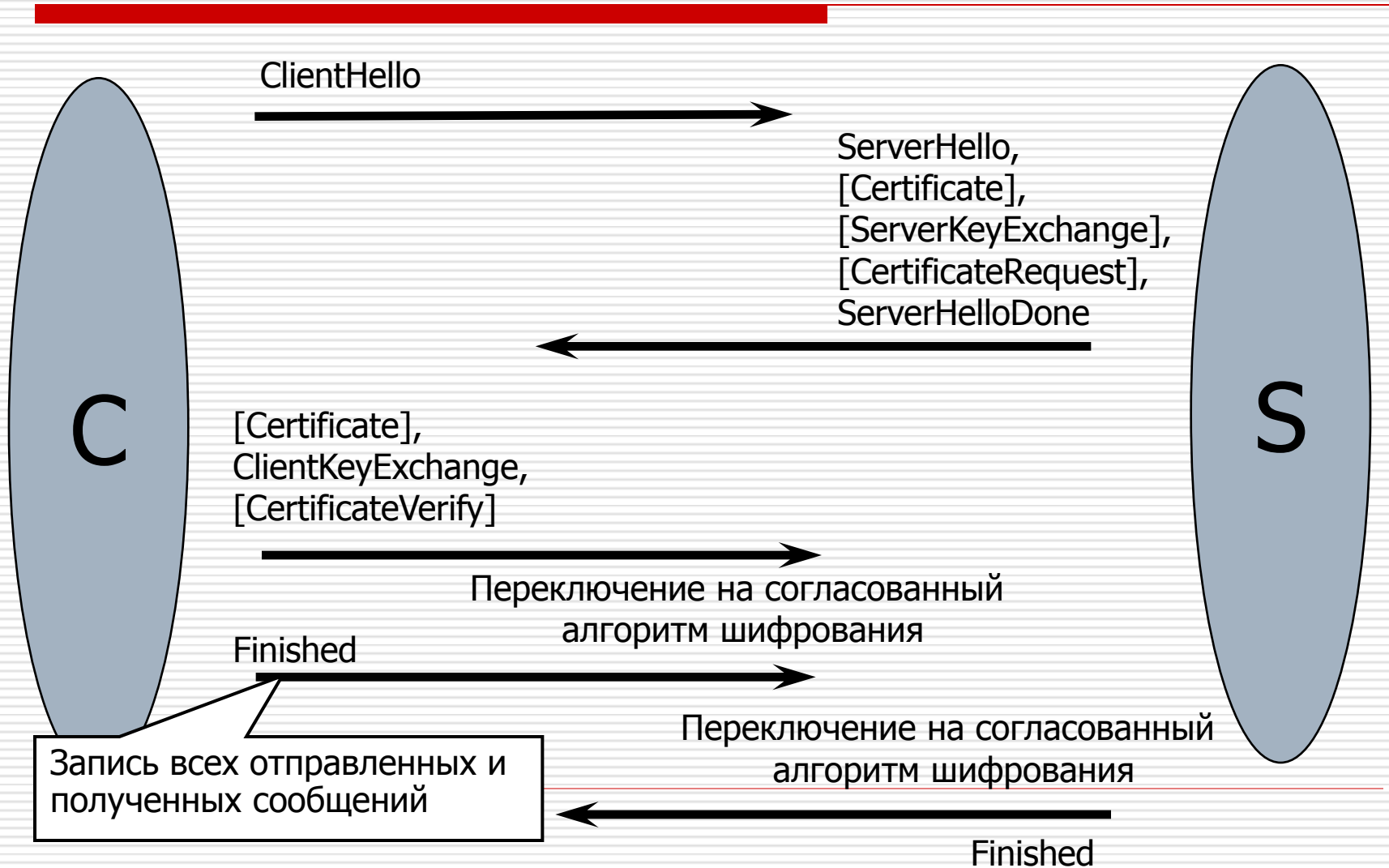
ОСНОВЫ SSL

- SSL состоит из двух протоколов
 - Протокол рукопожатия
 - Использует асимметрическую криптографию для установления общего секрета между клиентом и сервером
 - Протокол обмена сообщениями
 - Использует общий секрет, выработанный в первой фазе, для шифрования трафика между клиентом и сервером
-

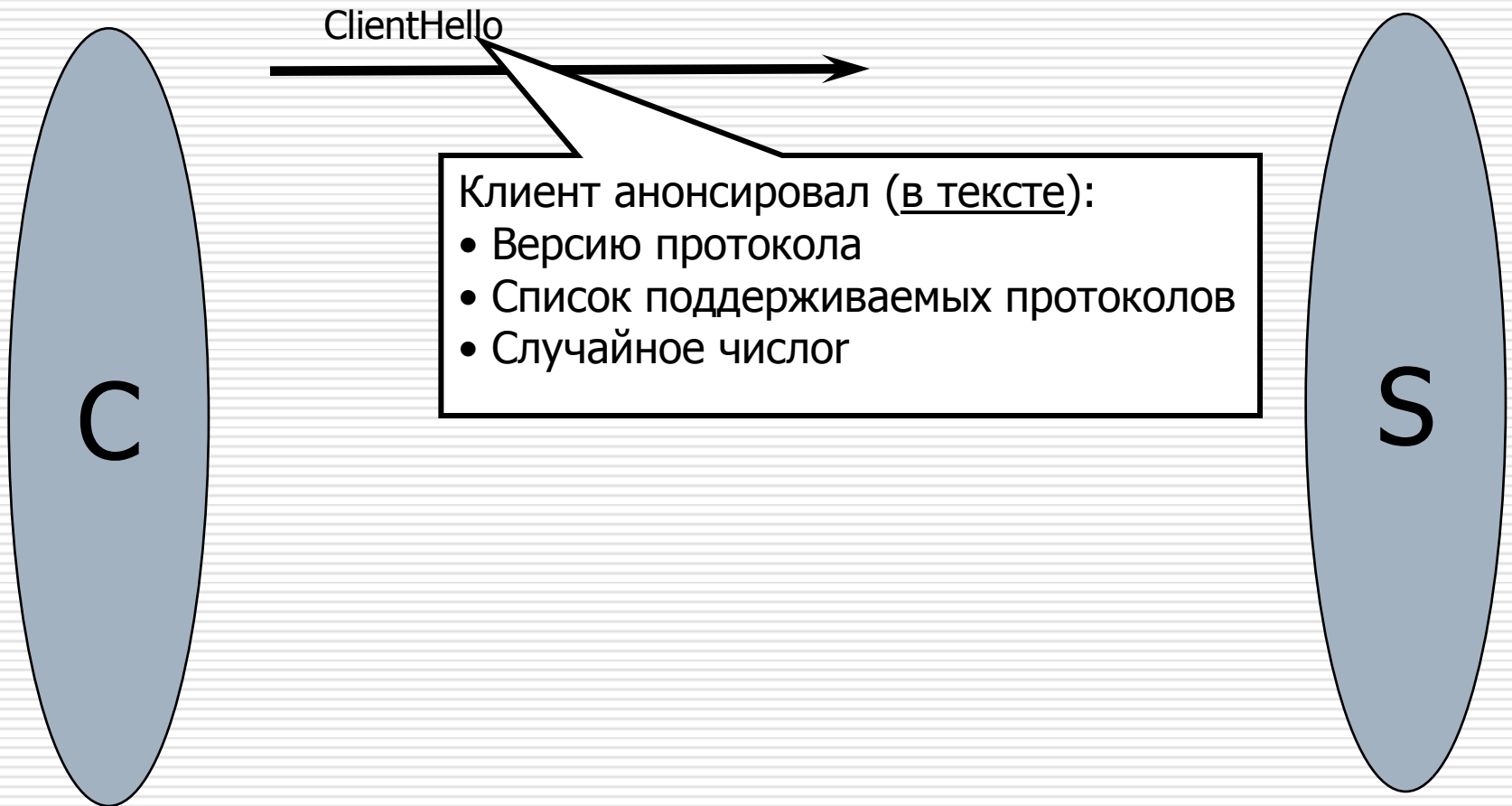
Протокол рукопожатия SSL

- Работает между клиентом (обычно, это веб-браузер) и сервером (возможно, это веб-сайт)
 - Согласование версии протокола и перечня криптографических алгоритмов
 - Аутентификация клиента и сервера (опционально)
 - Использует цифровые сертификаты чтобы узнавать публичные ключи друг друга и проверять подлинность
 - Использует публичные ключи для выработки общего секрета
-

Структура протокола рукопожатия



ClientHello



ClientHello (RFC)

```
struct {
```

```
    ProtocolVersion client_version;
```

Максимальная версия
протокола на клиенте

```
    Random random;
```

Session id (если клиент хочет
продолжить старую сессию)

```
    SessionID session_id;
```

```
    CipherSuite cipher_suites;
```

Список криптоалгоритмов на
клиенте (RSA, Diffie-Hellman, и
т.д.)

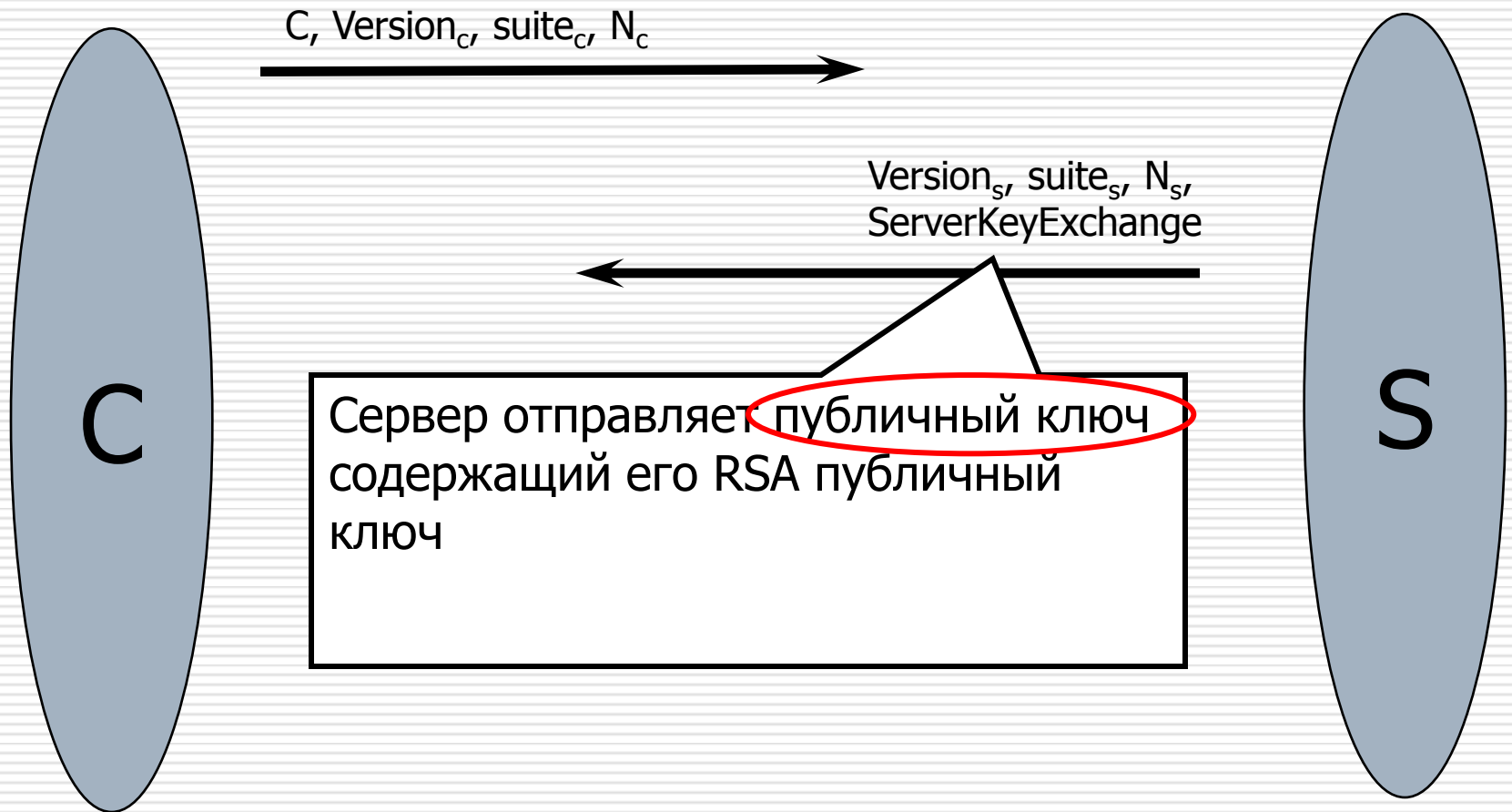
```
    CompressionMethod  
    compression_methods;
```

```
} ClientHello
```

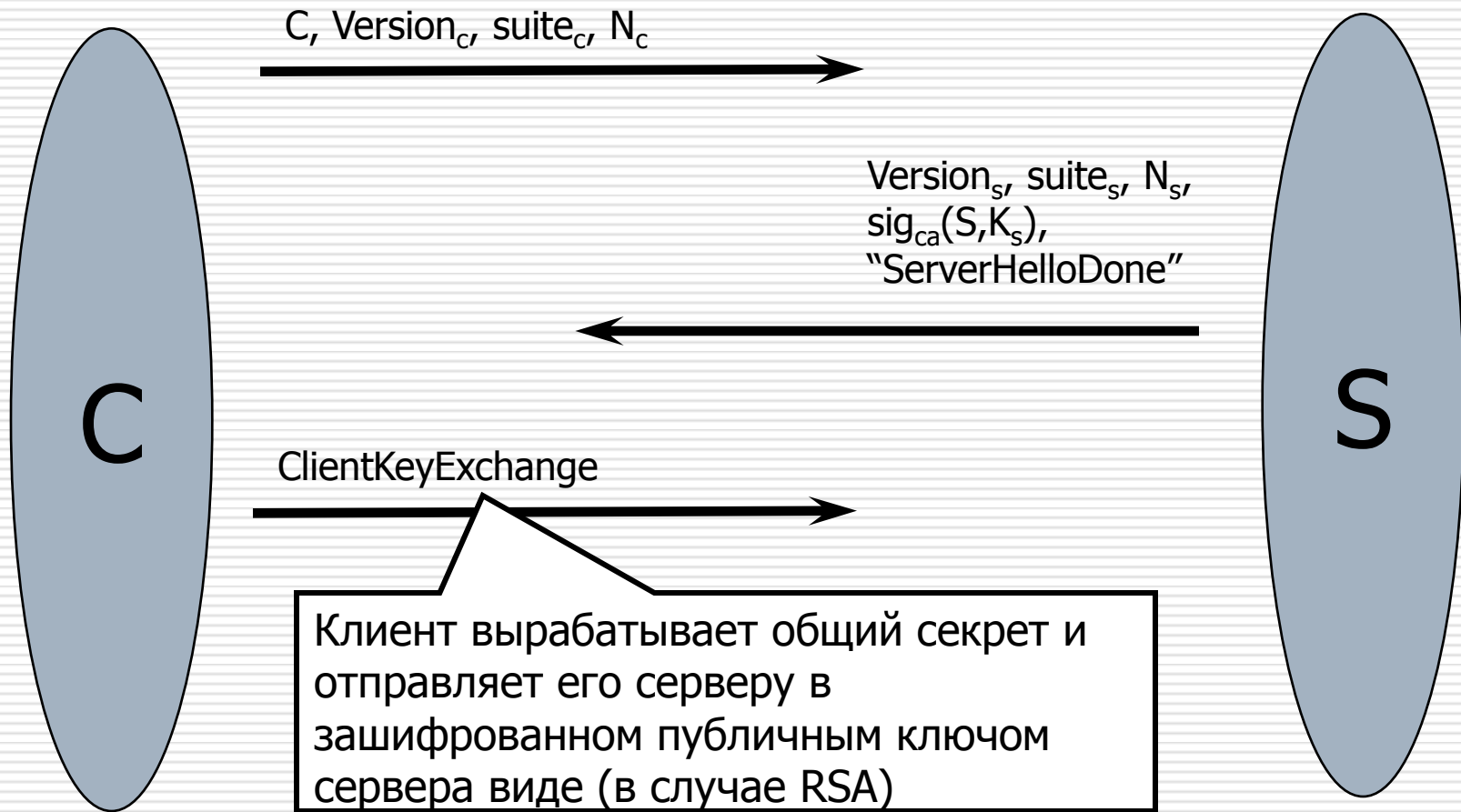
ServerHello



ServerKeyExchange



ClientKeyExchange



ClientKeyExchange (RFC)

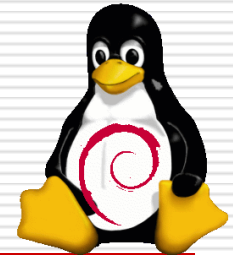
```
struct {  
    select (KeyExchangeAlgorithm) {  
        case rsa: EncryptedPreMasterSecret;  
        case diffie_hellman: ClientDiffieHellmanPublic;  
    } exchange_keys  
} ClientKeyExchange
```

```
struct {  
    ProtocolVersion client_version;  
    opaque random[46];  
} PreMasterSecret
```

Откуда берутся
случайные биты?

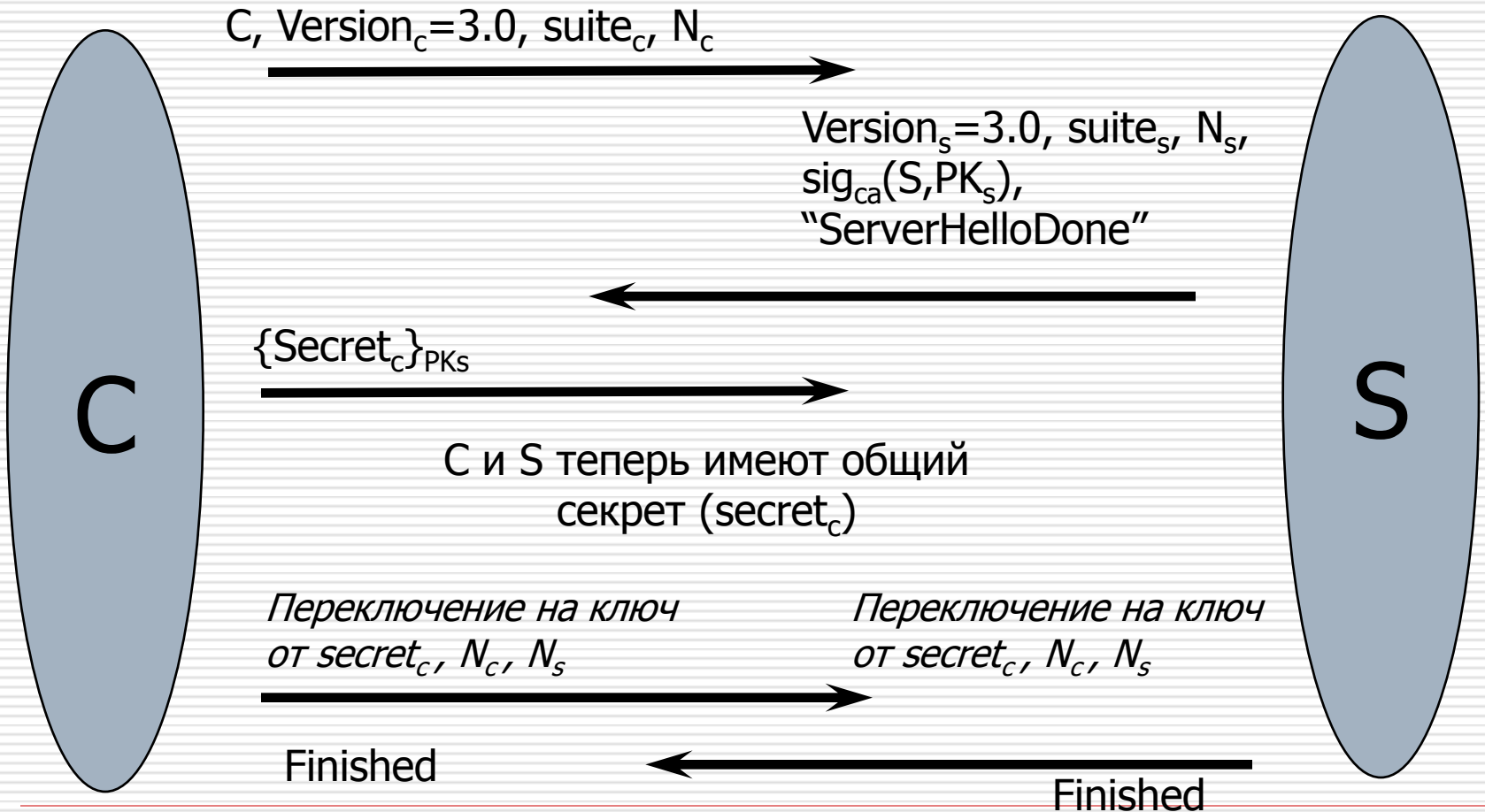
Случайные биты, из которых
будут сгенерены
симметричные ключи

Debian PRNG (2006-08)

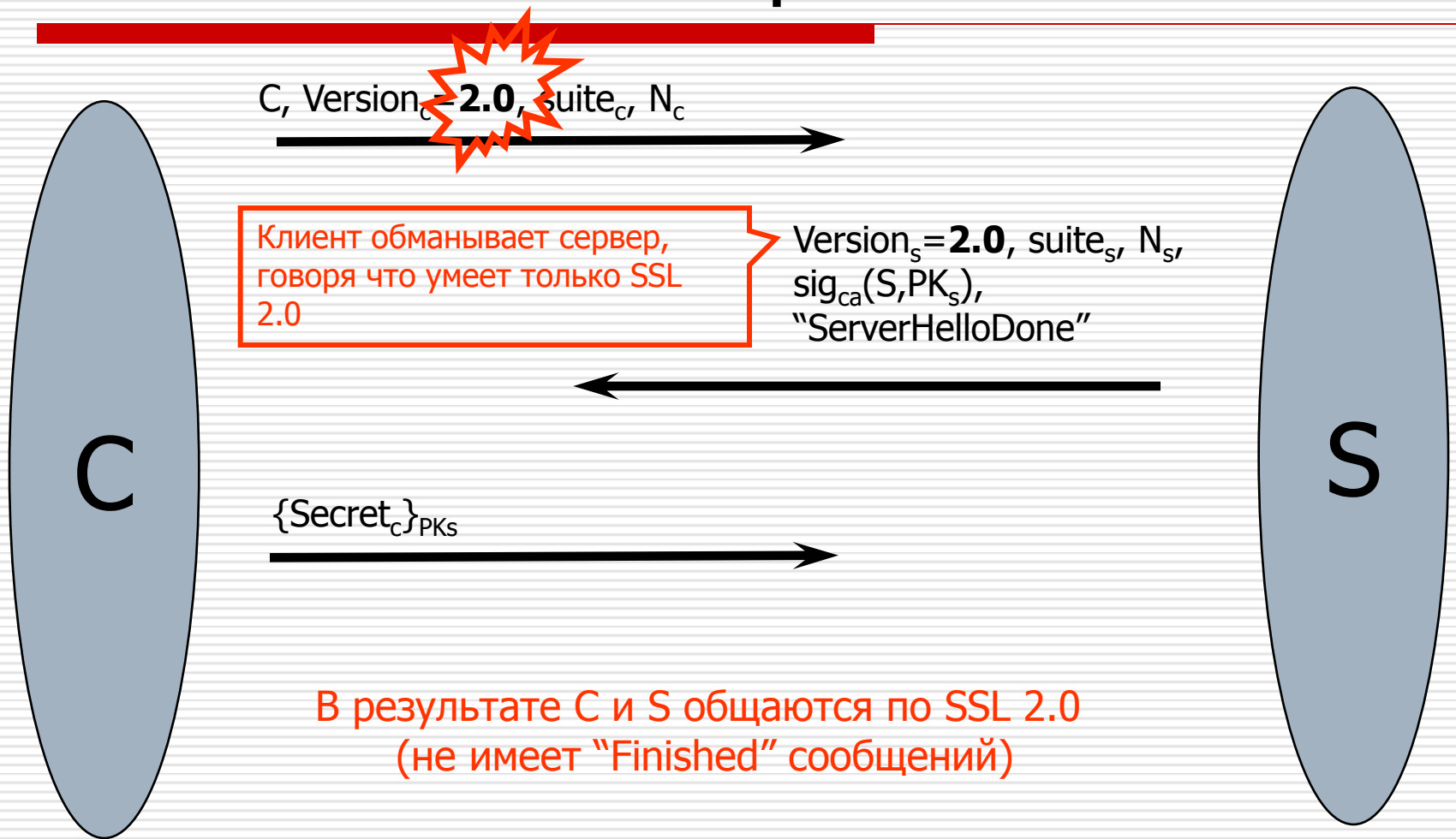


-
- Строчка кода в `md_rand` в Debian Linux
 - `MD_Update(&m,buf,j); /* purify complains */`
 - Без этой строчки сид для PRNG зависит только от ID процесса
 - Максимум на Linux по-умолчанию = 32768
 - Результат: все ключи, созданные с помощью Debian OpenSSL в 2006-08 сломаны
 - “Включая ключи SSH, OpenVPN, DNSSEC, ключевой материал сертификатов X.509 в SSL/TLS”
-

Рукопожатие в SSL 3.0



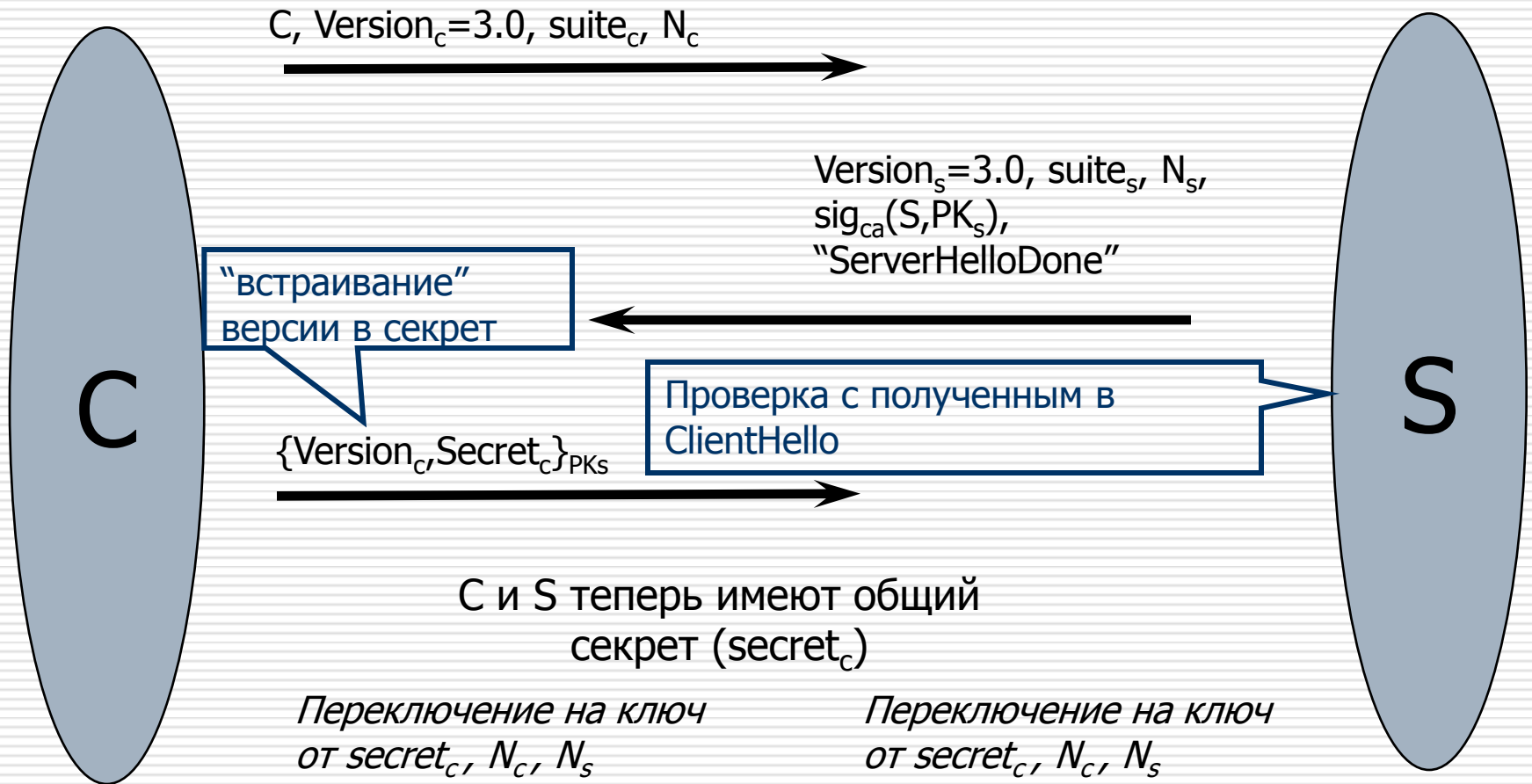
Атака отката версии



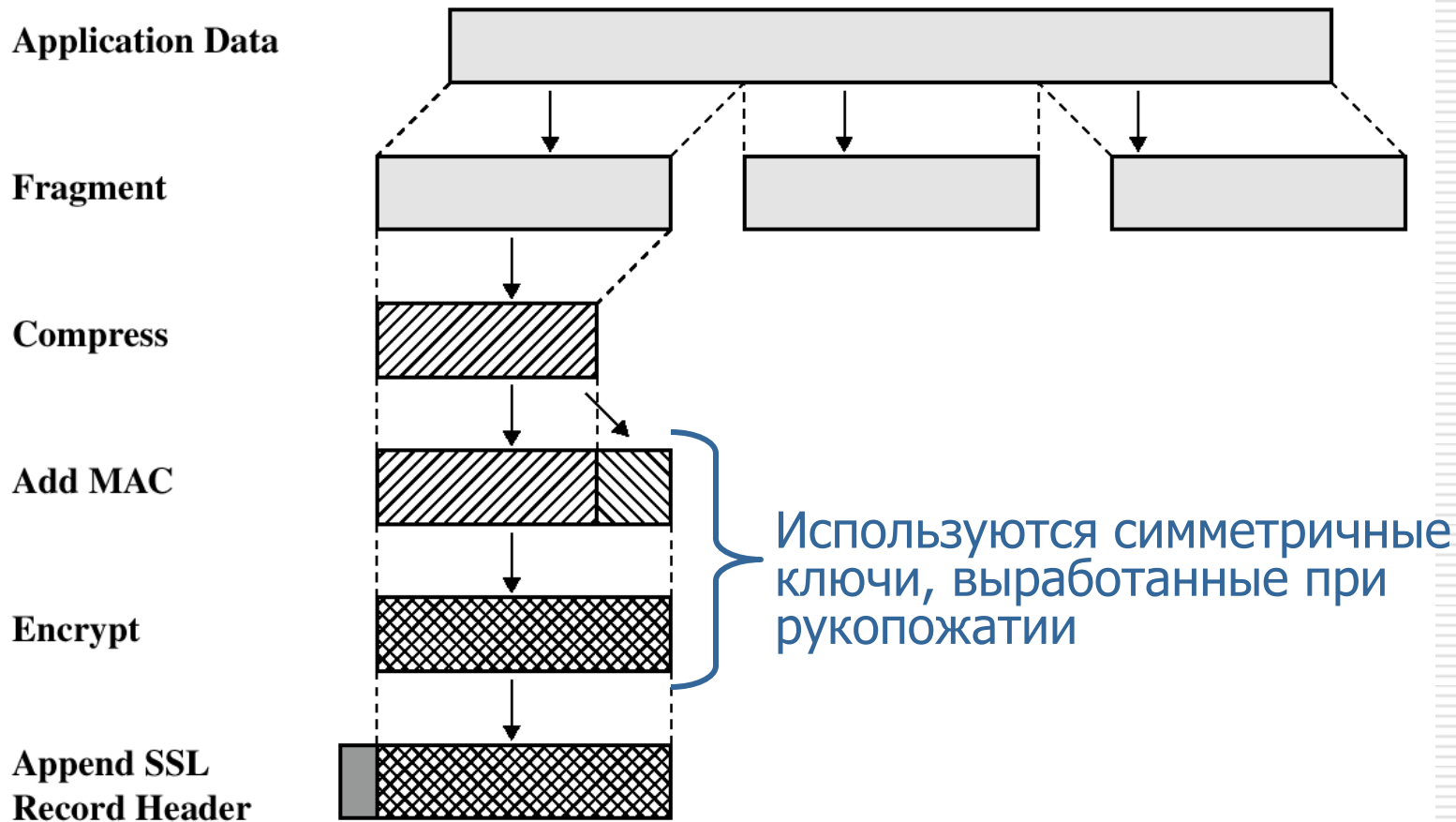
Уязвимости SSL 2.0

- ❑ Предпочтения криптоалгоритмов не аутентифицируются
 - Возможна атака “Cipher suite rollback”
 - ❑ Слабый MAC construction
 - ❑ SSL 2.0 использует паддинг при вычислении MAC, но поле длины паддинга не аутентифицировано
 - Атакующий может удалять байты в конце сообщения
 - ❑ MAC использует только 40 бит
 - ❑ Нет поддержки не-RSA алгоритмов
-

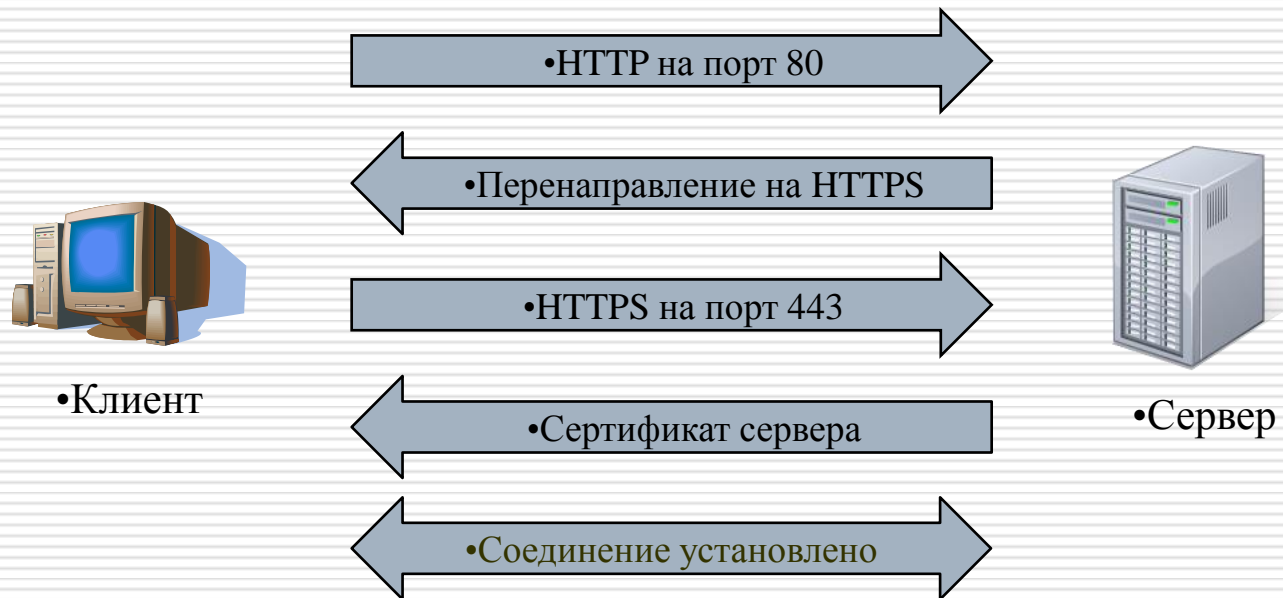
Проверка версии в SSL 3.0



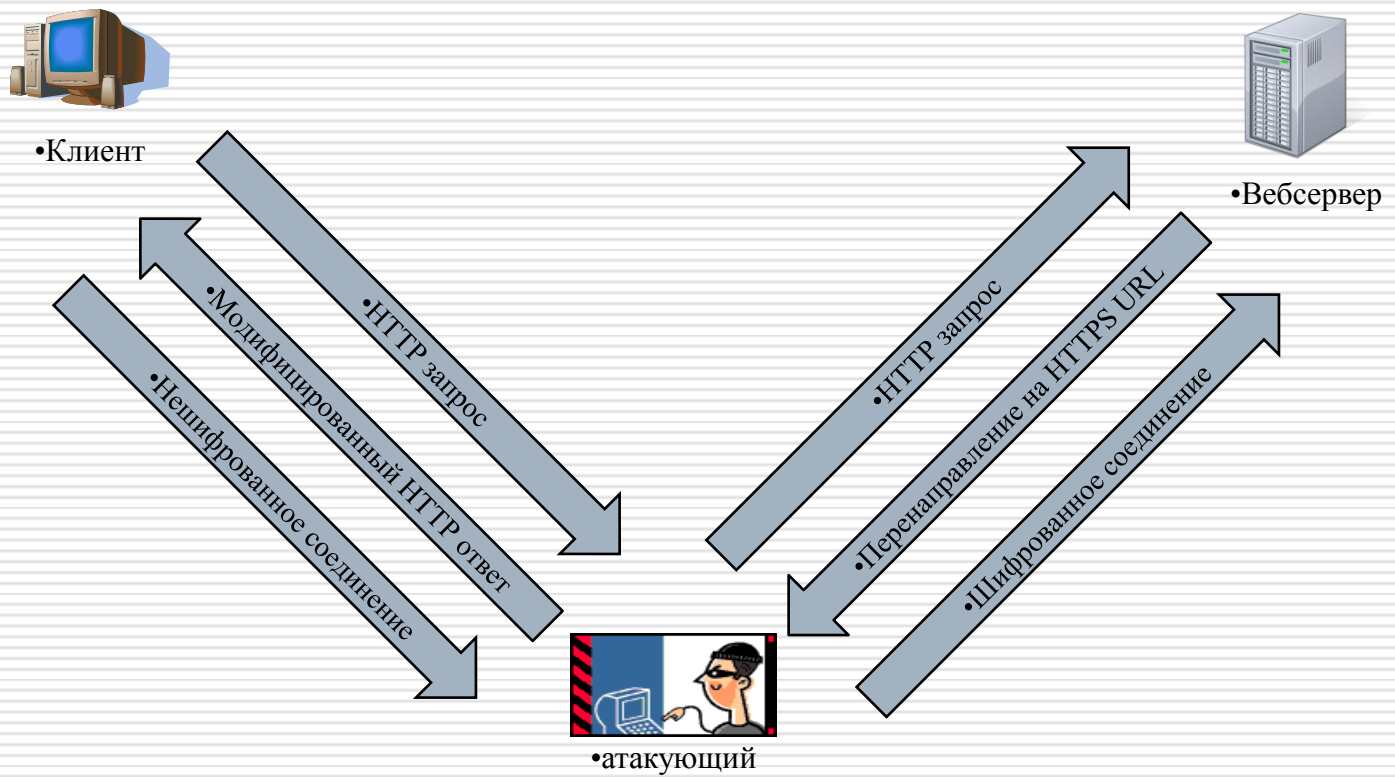
Защита сообщений в SSL/TLS



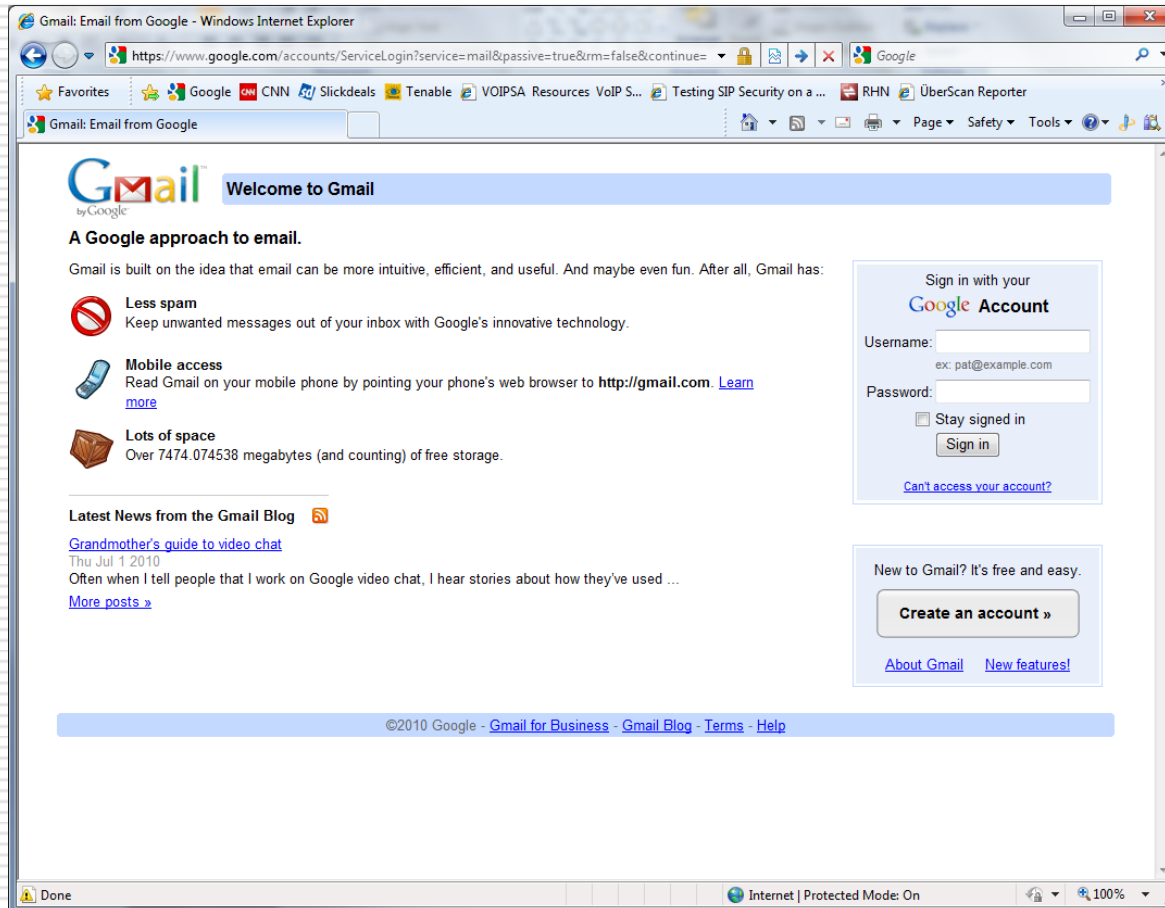
HTTP/HTTPS



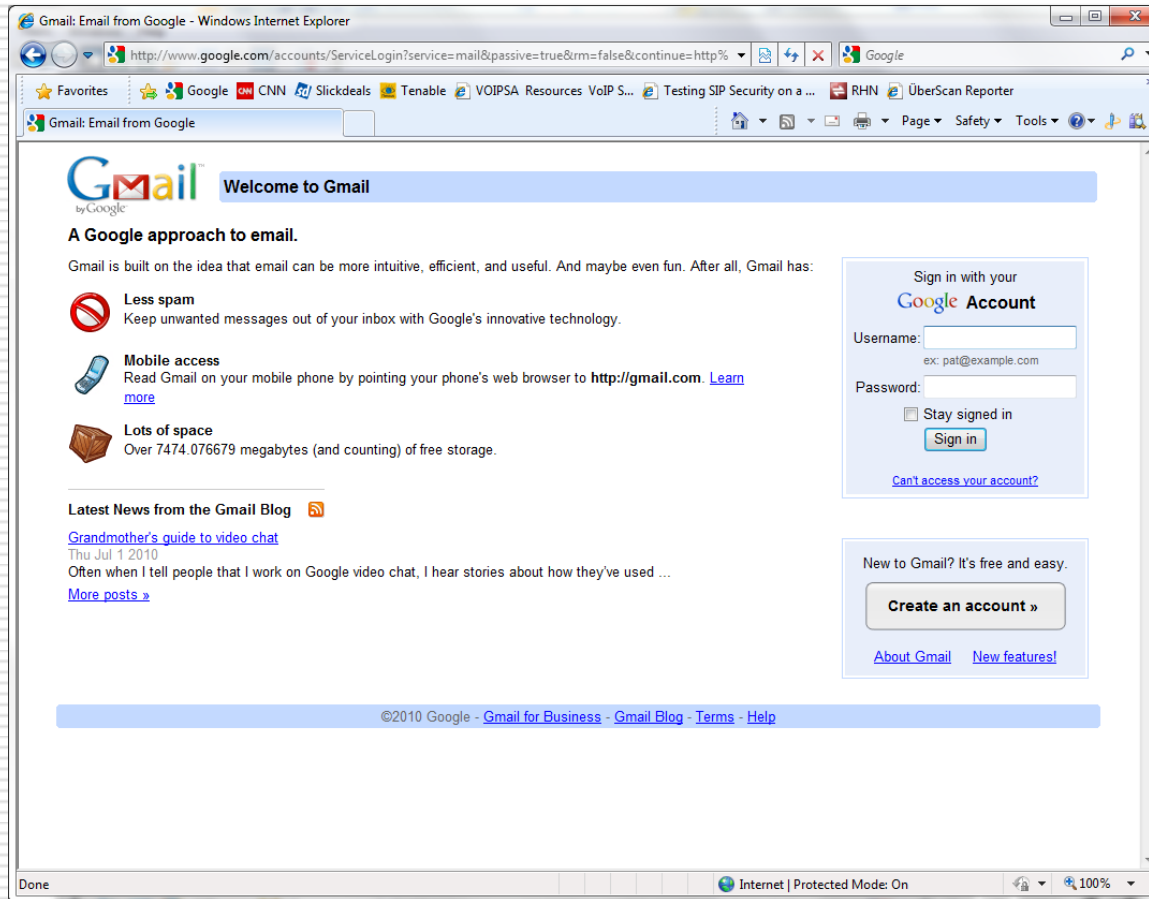
Hijacking Communication



Гмейл здорового человека

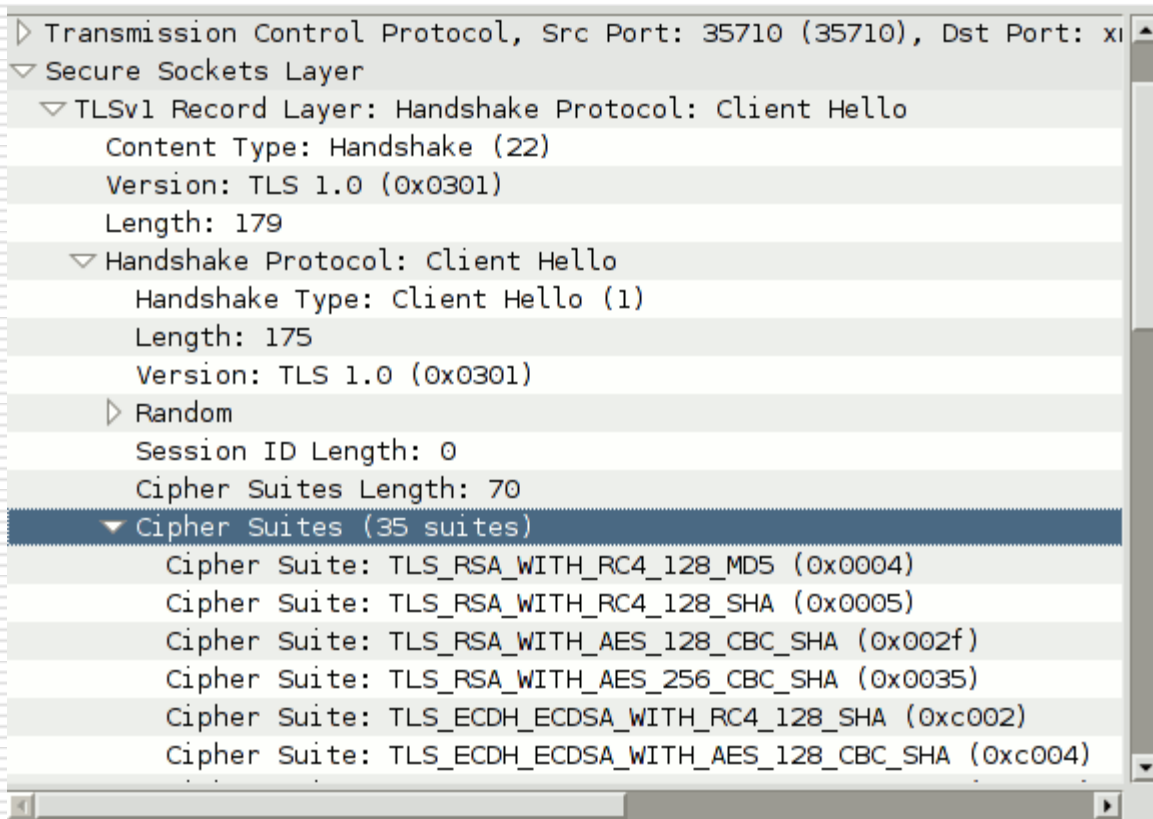


Гмейл курильщика



Ещё интересное


- http://op-co.de/blog/posts/android_ssl_downgrade/



GNU TLS: goto fail

- <http://www.gnutls.org/security.html#GNUTLS-SA-2014-2>
- <https://www.gitorious.org/gnutls/gnutls/commit/6aa26f78150ccbdf0aec1878a41c17c41d358a3b>

corrected return codes

 Nikos Mavrogiannopoulos 5 days ago

Summary

- lib/x509/verify.c (16) -----+-----

Inline diffs Side by side diffs Raw diff Raw patch

lib/x509/verify.c (+10/-6)

141	141	if (result < 0)
142	142	{
143	143	gnutls_assert ();
144	144	goto cleanup;
	144	goto fail;
145	145	}
146	146	
147	147	result =
...	...	
150	150	if (result < 0)
151	151	{
152	152	gnutls_assert ();
153	153	goto cleanup;
	153	goto fail;
154	154	}
155	155	
156	156	result =
...	...	
158	158	if (result < 0)
159	159	{
160	160	gnutls_assert ();
161	161	goto cleanup;
	161	goto fail;
162	162	}

Анализ HTTPS на pass.yandex.ru

https://www.ssllabs.com/ssltest/analyze.html?d=pass.yandex.ru

☆ ▾ ↻  Google



[Home](#) [Projects](#) [Qualys.com](#) [Contact](#)

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > pass.yandex.ru

SSL Report: pass.yandex.ru

Assessed on: Tue Mar 04 13:43:46 UTC 2014 | [Clear cache](#)

[Scan Another >>](#)

	Server	Domain(s)	Test time	Grade
1	213.180.204.51 Ready	pass.yandex.ru	Tue Mar 04 13:39:50 UTC 2014 Duration: 58.982 sec	A
2	213.180.193.51 pass.yandex.ru Ready	pass.yandex.ru	Tue Mar 04 13:40:49 UTC 2014 Duration: 59.58 sec	A
3	77.88.21.51 pass.yandex.ru Ready	pass.yandex.ru	Tue Mar 04 13:41:48 UTC 2014 Duration: 59.506 sec	A
4	87.250.250.51 pass.yandex.ru Ready	pass.yandex.ru	Tue Mar 04 13:42:47 UTC 2014 Duration: 59.73 sec	A

SSL Report: 4/20

Анализ HTTPS на login.mos.ru

https://www.ssllabs.com/ssltest/analyze.html?d=login.mos.ru

Google



Home Projects Qualys.com Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > login.mos.ru

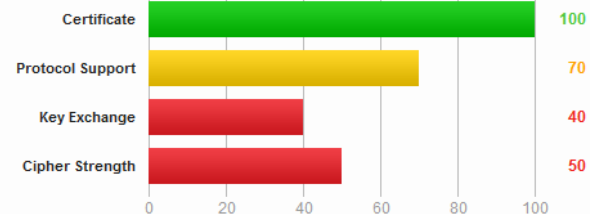
SSL Report: login.mos.ru (212.45.30.108)

Assessed on: Tue Mar 04 13:41:17 UTC 2014 | [Clear cache](#)

[Scan Another »](#)

Summary

Overall Rating



Documentation: [SSL/TLS Deployment Best Practices](#), [SSL Server Rating Guide](#), and [OpenSSL Cookbook](#).

The server supports only older protocols, but not the current best TLS 1.2. Grade capped to B.

There is no support for secure renegotiation. [MORE INFO »](#)

The server does not support Forward Secrecy with the reference browsers. [MORE INFO »](#)

Задание

- Настроить раздачу статической страницы с авторизацией по клиентскому SSL сертификату:
 - Nginx или apache
 - Генерация сертификатов – openssl + easy-rsa
 - Экспорт сертификатов в пригодном для браузеров виде
-