ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

ФАКУЛТЕТ ПО ТЕЛЕКОМУНИКАЦИИ

***Тема*: Протокол SSL и TSL. Принцип на работа. Метод за автентификация на сесията. Мтод за защита на данните в пакета. Режим на работа. Методи за конфигурироване в Linux.**

Изготвил : Данил Браснибрада
ФН 113317001

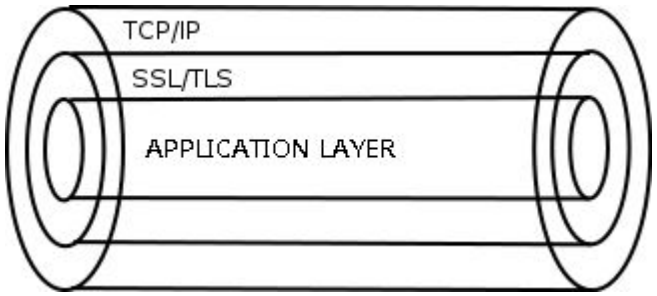
София 2018

SSL/TLS (Secure Sockets Layer)

- Криптографски протокол за връзка клиент-сървър, разработен от Netscape за пренасяне на информация през Интернет.
- Сигурност: Симетричното криптиране защитава предаваната информация от четене от неоторизирани лица.
- Автентификация: "Идентичността" на участника в връзката може да бъде проверена, като се използва асиметрично криптиране.
- Целостта: Всяко съобщение съдържа код за удостоверяване на съобщения (MAC), с който можете да проверите дали данните не са били променени или загубени по време на прехвърлянето.

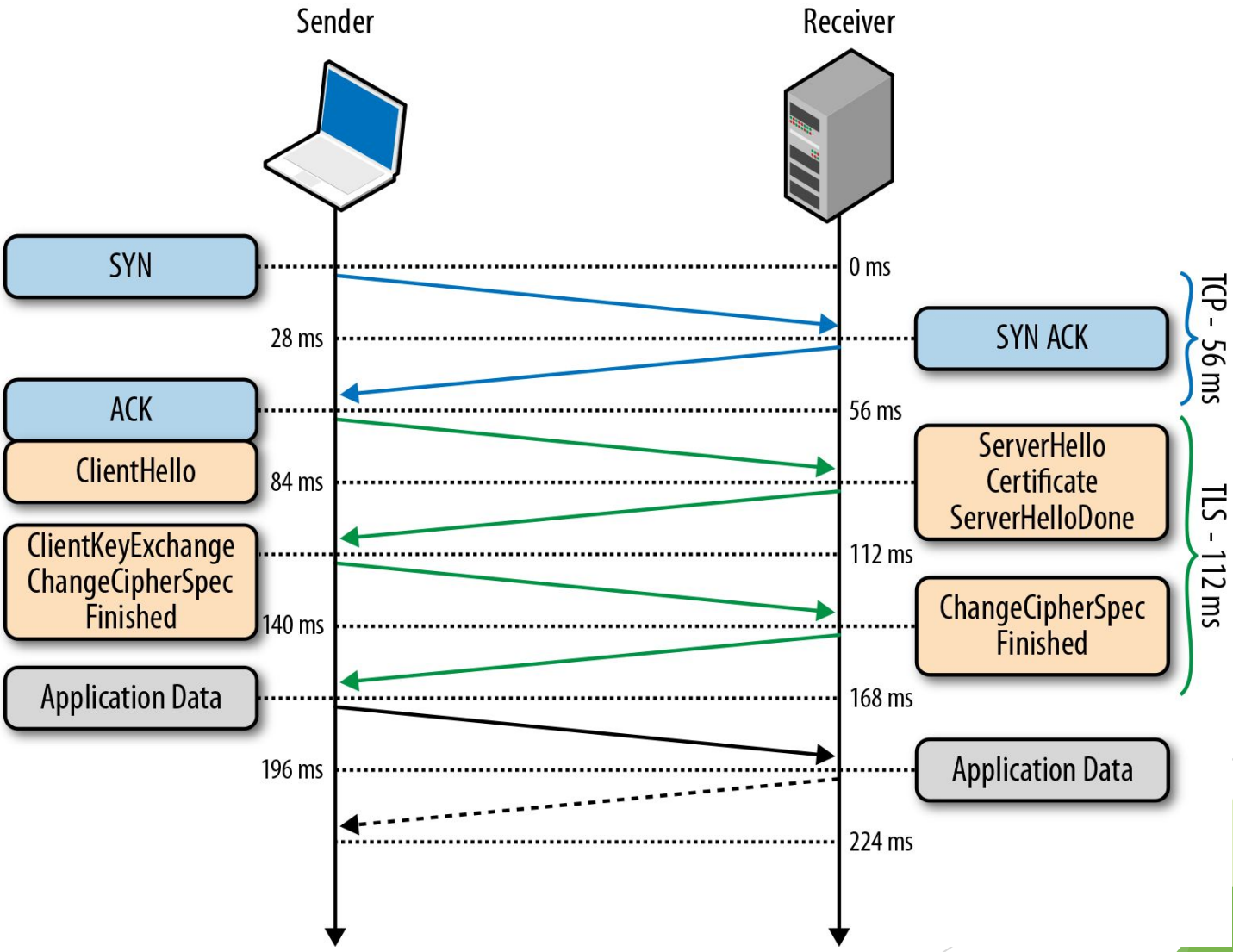
Принцип на работа SSL/TSL

- Над TCP / IP протокола се създава криптиран канал, в който данните се предават чрез протокола за приложения - HTTPS, FTPS и т.н. Ето как може да се представи гр



- Протоколът за приложение е "обвит" в TLS / SSL , а TLS/ SSL в TCP / IP. Всъщност данните за протокола на приложението се предават чрез TCP / IP, но те са криптирани.афично:

Автентификаиция на сесията



Автентификаиция на сесията

- Между клиента и сървъра се създава TCP връзка.
- След инсталирането на TCP, клиентът изпраща спецификация на сървъра (версията на протокола, която иска да използва, поддържани методи за шифроване и т.н.).
- Сървърът одобрява версията на използвания протокол, избира метода на шифроване от предоставения списък, връща своя сертификат и изпраща отговор на клиента (при желание сървърът може да поиска клиентски сертификат).
- В този момент протоколната версия и методът на шифроване се считат за одобрени, клиентът проверява подадения сертификат и инициира RSA или Diffie-Hellman, в зависимост от зададените параметри.
- Сървърът обработва изпратеното от клиента съобщение, проверява MAC и изпраща на клиента окончателно (“Finished") съобщение в шифрована форма.
- Клиентът декриптира полученото съобщение, проверява MAC и ако всичко е наред, връзката се установява и започва обменът на данни за приложенията.

RSA Генериране на ключа

- Всеки потребител генерира двойка публичен и частен ключ, така:
- Избират се две големи пройзволни прости числа **p, q**
- Изчислява се тяхното произведение **N=p.q**
- Изчислява се Функция на Ойлера $\Phi(N) = (p-1)(q-1)$
- Избира се пройзволно e – криптраща експонента, такава, че $1<e$

  Да няма най-малко общ делител с $\Phi(N)$ $\gcd(e,\phi(N))=1$
- Решава се уравнението за да се намери декриптиращия ключ d

$$d = \frac{k*\Phi(N)+1}{e}, \quad 0 \le d \le N$$

- Обявява се техния публичен ключ: $K_{пуб}=\{e,N\}$
- Запазва се скрит частния ключ: $K_{секр}=\{d,p,q\}$
- За да се криптира съобщението M изпращащия:

  Взема публичния ключ на получателя $K_{пуб}=\{e,N\}$

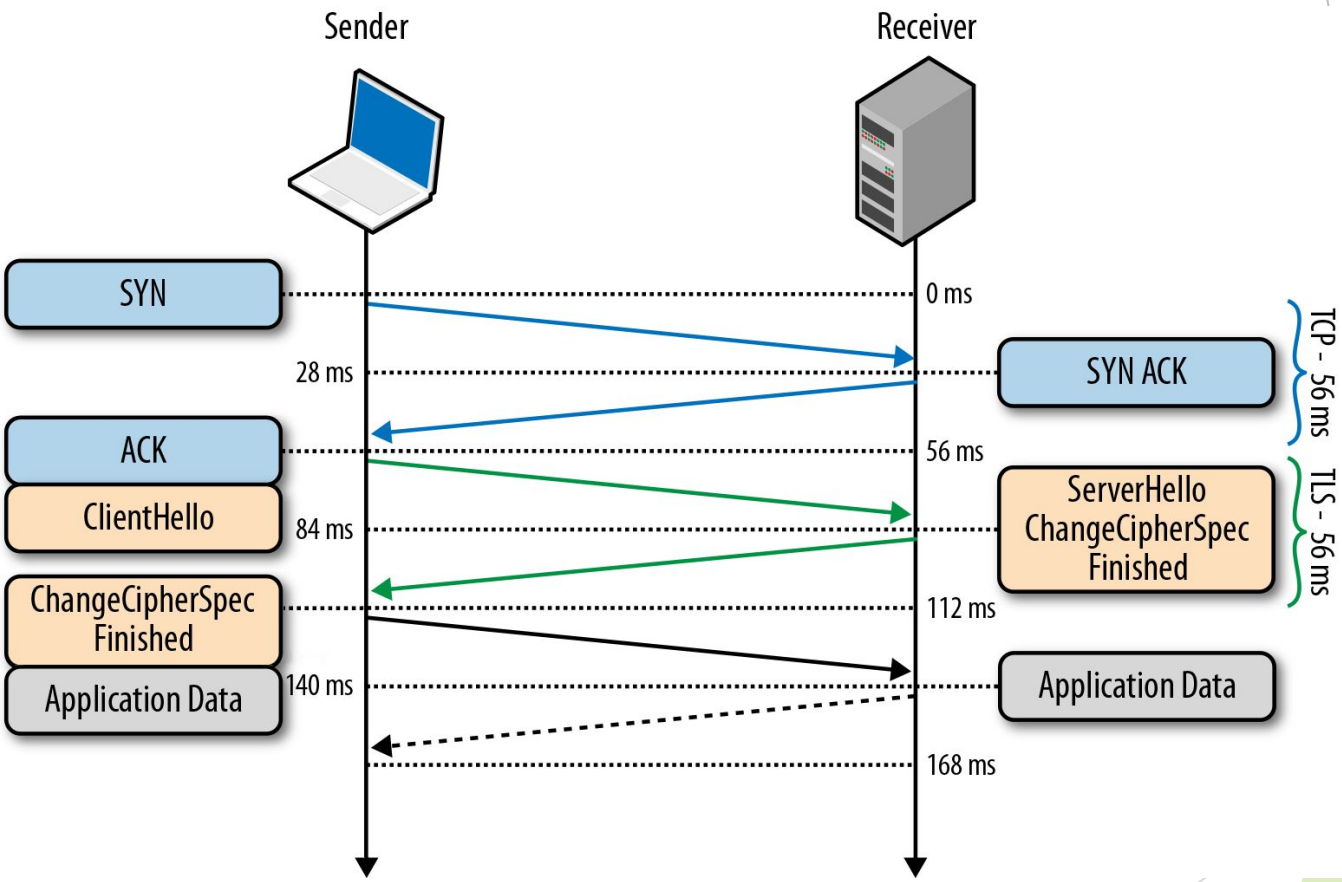
  Изчислява: $C=M^e \bmod N$, където $0 \le M$

RSA Генериране на ключа

- Всеки потребител генерира двойка публичен и частен ключ, така:
- Избират се две големи пройзволни прости числа **p, q**
- Изчислява се тяхното произведение **N=p.q**
- Изчислява се Функция на Ойлера $\Phi(N) = (p-1)(q-1)$
- Избира се пройзволно e – криптраща експонента, такава, че $1<e$
  Да няма най-малко общ делител с $\Phi(N)$ $\gcd(e,\phi(N))=1$
- Решава се уравнението за да се намери декриптиращия ключ d

Using versions of SSL\TSL		
Version	Security	% of websites in the World
SSL 2.0	NO	4.9 %
SSL 3.0	NO	16.6 %
TLS 1.0	Maybe	94.7 %
TLS 1.1	YES	82.6 %
TLS 1.2		85.5 %

Abbreviated handshake



Abbreviated handshake

TLS Handshake е доста дълъг и скъп от гледна точка на изчислителните разходи. Поради това е разработена процедура, която ви позволява да възобновите предишната прекъсната връзка въз основа на вече конфигурираните данни.

- Първото съобщение ServerHello, сървърът генерира и изпраща идентификатор за сесия от 32 байта до клиента.
- Клиентът съхранява изпратения идентификатор и го включва (разбира се, ако съществува) в оригиналното съобщение ClientHello.
- Ако и клиентът, и сървърът имат идентични идентификатори на сесии, общата връзка се установява

Конфигуриране SSL\TSL в Linux.

```
Version: 1.11.6
OpenSSL 1.0.2h  3 May 2016

Testing SSL server sergeysl.ru on port 443

  TLS renegotiation:
Secure session renegotiation supported

  TLS Compression:
Compression disabled

  Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.2   256 bits   ECDHE-RSA-AES256-GCM-SHA384     Curve P-256 DHE 256
Accepted  TLSv1.2   128 bits   ECDHE-RSA-AES128-GCM-SHA256     Curve P-256 DHE 256
Accepted  TLSv1.2   256 bits   ECDHE-RSA-AES256-SHA384         Curve P-256 DHE 256
Accepted  TLSv1.2   256 bits   ECDHE-RSA-AES256-SHA            Curve P-256 DHE 256
Accepted  TLSv1.2   128 bits   ECDHE-RSA-AES128-SHA256         Curve P-256 DHE 256
Accepted  TLSv1.2   128 bits   ECDHE-RSA-AES128-SHA            Curve P-256 DHE 256
Accepted  TLSv1.2   112 bits   ECDHE-RSA-DES-CBC3-SHA          Curve P-256 DHE 256
Accepted  TLSv1.2   256 bits   AES256-GCM-SHA384
Accepted  TLSv1.2   128 bits   AES128-GCM-SHA256
Accepted  TLSv1.2   256 bits   AES256-SHA256
Accepted  TLSv1.2   128 bits   AES128-SHA256
Accepted  TLSv1.2   256 bits   AES256-SHA
Accepted  TLSv1.2   128 bits   AES128-SHA
Accepted  TLSv1.2   112 bits   DES-CBC3-SHA
Accepted  TLSv1.2   256 bits   CAMELLIA256-SHA
Accepted  TLSv1.2   128 bits   CAMELLIA128-SHA
Preferred TLSv1.1   256 bits   ECDHE-RSA-AES256-SHA            Curve P-256 DHE 256
Accepted  TLSv1.1   128 bits   ECDHE-RSA-AES128-SHA            Curve P-256 DHE 256
Accepted  TLSv1.1   112 bits   ECDHE-RSA-DES-CBC3-SHA          Curve P-256 DHE 256
Accepted  TLSv1.1   256 bits   AES256-SHA
Accepted  TLSv1.1   128 bits   AES128-SHA
Accepted  TLSv1.1   112 bits   DES-CBC3-SHA
Accepted  TLSv1.1   256 bits   CAMELLIA256-SHA
Accepted  TLSv1.1   128 bits   CAMELLIA128-SHA
Preferred TLSv1.0   256 bits   ECDHE-RSA-AES256-SHA            Curve P-256 DHE 256
Accepted  TLSv1.0   128 bits   ECDHE-RSA-AES128-SHA            Curve P-256 DHE 256
Accepted  TLSv1.0   112 bits   ECDHE-RSA-DES-CBC3-SHA          Curve P-256 DHE 256
Accepted  TLSv1.0   256 bits   AES256-SHA
Accepted  TLSv1.0   128 bits   AES128-SHA
Accepted  TLSv1.0   112 bits   DES-CBC3-SHA
Accepted  TLSv1.0   256 bits   CAMELLIA256-SHA
Accepted  TLSv1.0   128 bits   CAMELLIA128-SHA

  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  sergeysl.ru
Altnames: DNS:sergeysl.ru
Issuer:   StartCom Class 1 DV Server CA

Not valid before: Jun 10 08:25:47 2016 GMT
Not valid after:  Jun 10 08:25:47 2017 GMT
```

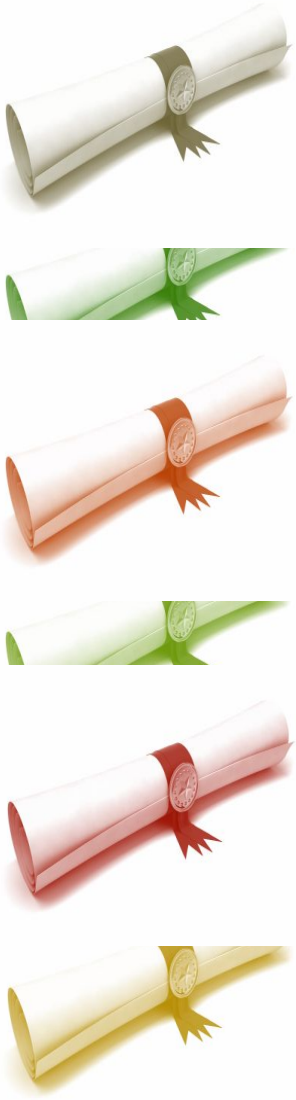
```
  TLS renegotiation:
Secure session renegotiation supported

  TLS Compression:
Compression disabled

  Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.0  256 bits  DHE-RSA-AES256-SHA              DHE 1024 bits
Accepted  TLSv1.0  256 bits  DHE-RSA-CAMELLIA256-SHA        DHE 1024 bits
Accepted  TLSv1.0  256 bits  ADH-AES256-SHA                 DHE 1024 bits
Accepted  TLSv1.0  256 bits  ADH-CAMELLIA256-SHA           DHE 1024 bits
Accepted  TLSv1.0  256 bits  AES256-SHA
Accepted  TLSv1.0  256 bits  CAMELLIA256-SHA
Accepted  TLSv1.0  128 bits  DHE-RSA-AES128-SHA              DHE 1024 bits
Accepted  TLSv1.0  128 bits  DHE-RSA-CAMELLIA128-SHA        DHE 1024 bits
Accepted  TLSv1.0  128 bits  ADH-AES128-SHA                 DHE 1024 bits
Accepted  TLSv1.0  128 bits  ADH-CAMELLIA128-SHA           DHE 1024 bits
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  128 bits  CAMELLIA128-SHA
Accepted  TLSv1.0  128 bits  ADH-RC4-MD5                   DHE 1024 bits
Accepted  TLSv1.0  128 bits  RC4-SHA
Accepted  TLSv1.0  128 bits  RC4-MD5
Accepted  TLSv1.0  112 bits  EDH-RSA-DES-CBC3-SHA          DHE 1024 bits
Accepted  TLSv1.0  112 bits  ADH-DES-CBC3-SHA             DHE 1024 bits
Accepted  TLSv1.0  112 bits  DES-CBC3-SHA

  SSL Certificate:
Signature Algorithm: md5WithRSAEncryption
RSA Key Strength:    1024

Subject:  mail.[redacted].ru
Issuer:   mail.[redacted].ru

Not valid before: Aug 24 11:02:28 2005 GMT
Not valid after:  Aug 22 11:02:28 2015 GMT
```

SSL / TLS приложения

В стандартните приложения е възможно да се разпредели отделен порт за организиране на сигурна SSL / TLS връзка (например POP3S-995, IMAPS-993 HTTPS-443) или използване на същия порт с възможност за превключване към защитена връзка посредством командата START TLS (например ESMTP)

- HTTPS
- POP3S, IMAPS
- ESMTP
- stunnel

Сертификати



DV сертификат: удостоверява само домейна

OV сертификат: удостоверява домейна и организацията

EV сертификат: Разширена организационна проверка

SAN сертификат: удостоверява множество домейни

WILDCARD: сертифицира поддомейни на едно ниво

CODE SIGNING Приложено за разработчики

SSL и TSL алгоритми

Алгоритми за обмен на ключове:

- RSA (Ron Rivest, Adi Shamir, Leonard Adleman – MIT, 1977)
- Diffie-Hellman (Whitfield Diffie, Martin Hellman / Ralph Merkle – 1976)
- DSA (Digital Signature Algorithm / David W. Kravitz – 1991)
- SRP (Secure Remote Password Protocol)
- PSK (Pre-shared key)

Симетрични крипто алгоритми:

- RC4™ (Ron Rivest/RSA Security – 1987) или ARCFOUR (1994)
- 3DES (Triple Data Encryption Standard – IBM, 1973-74)
- AES (Advanced Encryption Standard AKA “Rijndael” – Joan Daemen and Vincent Rijmen ~1997)
- Camellia (European Union's NESSIE project, Japanese CRYPTREC project – Mitsubishi & NTT, 2000)
- IDEA™ (International Data Encryption Algorithm – Xuejia Lai&James Massey/ ETH Zurich, 1991)

Алгоритми за хеширане:

- HMAC-MD5 (Message-Digest algorithm 5 – Ron Rivest, 1991)
- HMAC-SHA (Secure Hash Algorithm, 1993)

Благодаря ви за вниманието!!!