Программная инженерия

Москва
2026

Лекция № 5. Шифрование

Основы кибербезопасности

Белявский Д.А.

Результаты контрольной работы по инцидентам

Турнирная таблица по командам

1	girls	Noname-II (249)	Водолазы	Пингвины	Синий кит	30
2	Cherrypickme	Noname-I (249)	Noname-III (248)	Дети Эпштейна	Валюнчики и Саша	29
	Добряки	Сила Сибири	Тимур и его команда	Факультет крепких напитков		
3	•0o_Уцуцуга_o0•	Team BBLU	Даниил Колбасенко	Капибара	Опоздавшие	28
4	Noname-IV (247)	Десептиконы	Терафлопсы	Крайний из армян		27
	Уточки	Никита Тищенко и Ко.				
5	4 энергоблок	Чемпионы				26


Криптография

Что такое криптография?

Наука о безопасном хранении и передаче информации с помощью сокрытия информация до «степени» неузнаваемости (шифрования)

Один из самых ранних методов шифрования – **шифр Цезаря**

Текст: ДА ПРЕБУДЕТ С НАМИ СИЛА

Шифр-текст: ЗД УФИЕЧЗИЦ Х СДРМ ХМПД

Ключ шифрования: 4

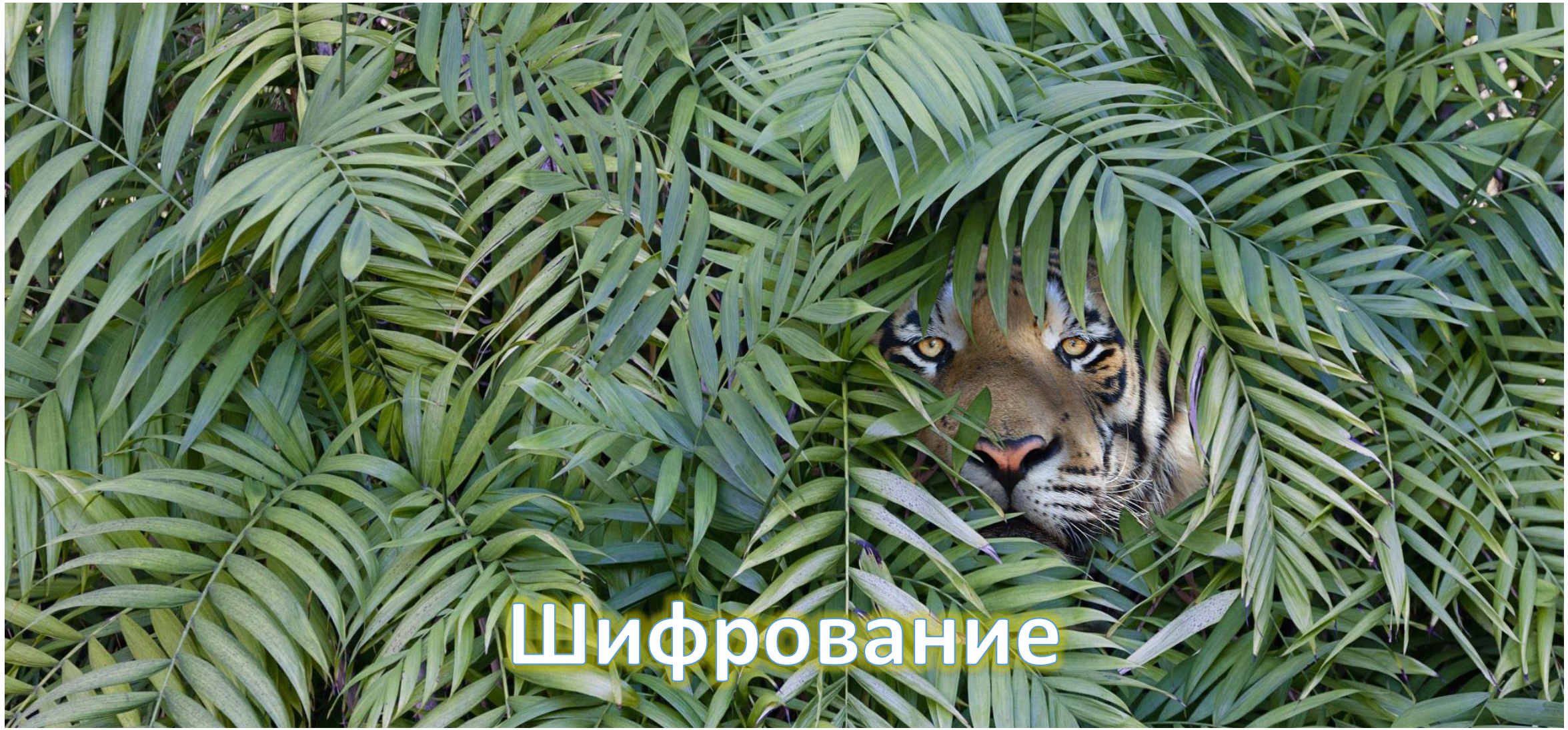
Пробуем расшифровать шифр Цезаря

Шифр-текст: ЩЫЭЫФ Х ЮЫШЪГС РСЪИ ДАРСЮЪЗЦ

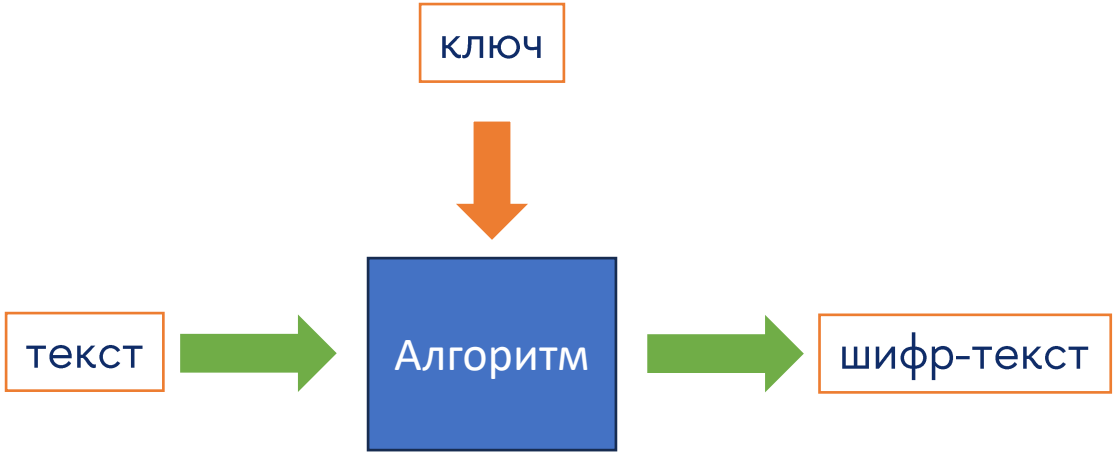
Ключ шифрования: 13

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л

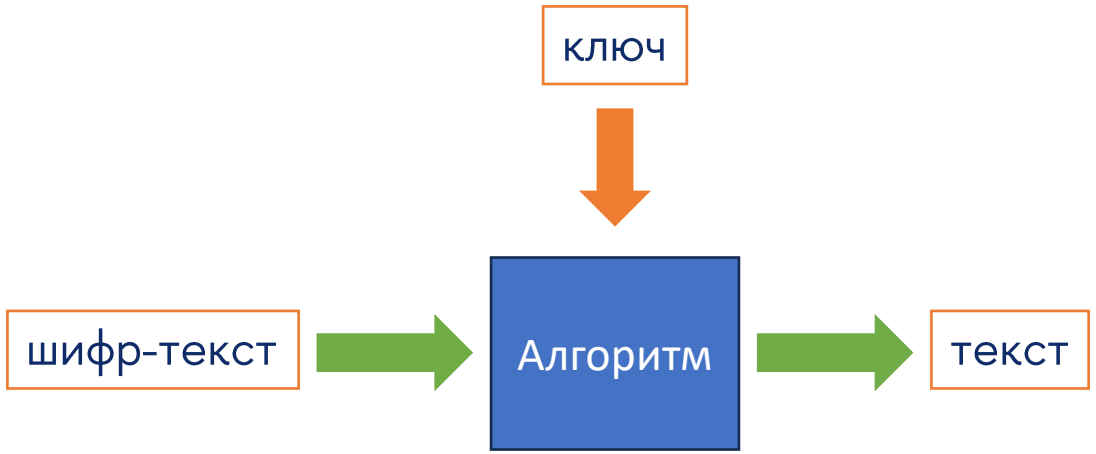
Текст: МОРОЗ И СОЛНЦЕ ДЕНЬ ЧУДЕСНЫЙ


Шифрование

Шифрование



Расшифрование



Симметричное шифрование

Текст: ПРИВЕТ_КИБЕР-МИР

Простая замена (перестановка) по столбцам

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

П	Е	И	-
Р	Т	Б	М
И	_	Е	И
В	К	Р	Р

Ключ шифрования:

Таблица 4x4

Шифр-текст:

ПЕИ-РТБМИ_ЕИВКРР

Простая замена (перестановка) с магическими квадратами

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Р	И	Р	-
Е	Б	Е	К
И	Т	_	Р
В	И	М	П

Ключ шифрования:

{16,3,2,13,5,10,11,8,…}

Шифр-текст:

РИР-ЕБЕКИТ_РВИМП

Симметричное шифрование

Используется ОДИНАКОВЫЙ ключ для шифрования и расшифрования

DES
(Data Encryption Standard)
Размер блока: 64 бит
Размер ключа: 64 бит

AES
(Advanced Encryption Standard)
Размер блока: 128 бит
Размер ключа: 128/192/256 бит

Camellia
Японский стандарт
Размер блока: 128 бит
Размер ключа: 128 бит

Кузнечик
ГОСТ 34.12-2018
Размер блока: 128 бит
Размер ключа: 256 бит

Магма
ГОСТ 34.12-2018
Размер блока: 64 бит
Размер ключа: 256 бит

Алгоритмы достаточно быстрые, что позволяет их широко применять

Небезопасная передача ключей шифрования

Режимы симметричного шифрования

Простая замена	ECB	Режим электронной шифровальной книги Electronic Codebook (ECB): Открытый текст обрабатывается блоками и каждый блок шифруется с одним и тем же ключом
Простая замена с зацеплением	CBC	Режим сцепления шифрованных блоков Cipher Block Chaining (CBC): В режиме CBC входной блок данных для алгоритма шифрования вычисляется как результат операции XOR текущего блока открытого текста и блока шифрованного текста, полученного на предыдущем шаге
Гаммирование с обратной связью	CFB	Режим шифрованной обратной связи Cipher Feedback (CFB): Полученный на предыдущем шаге шифрованный текст используется как входные данные для алгоритма шифрования с целью получения псевдослучайной последовательности (ПСП), XOR-разница которой и блока открытого текста определяет очередной блок шифрованного текста
Гаммирование	OFB	Режим обратной связи по выходу Output Feedback (OFB): Работает подобно CFB, но в качестве входных данных для алгоритма шифрования используются ранее полученные выходные данные

Примеры алгоритмов шифрования OpenSSL

```
Cipher Types
-aes-128-cbc               -aes-128-cbc-hmac-sha1     -aes-128-cbc-hmac-sha256  -aes-128-ccm               -aes-128-cfb
-aes-128-cfb1              -aes-128-cfb8              -aes-128-ctr              -aes-128-ecb               -aes-128-gcm
-aes-128-ofb               -aes-128-xts               -aes-192-cbc              -aes-192-ccm               -aes-192-cfb
-aes-192-cfb1              -aes-192-cfb8              -aes-192-ctr              -aes-192-ecb               -aes-192-gcm
-aes-192-ofb               -aes-256-cbc               -aes-256-cbc-hmac-sha1    -aes-256-cbc-hmac-sha256   -aes-256-ccm
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8             -aes-256-ctr               -aes-256-ecb
-aes-256-gcm               -aes-256-ofb               -aes-256-xts              -aes128                    -aes192
-aes256                    -bf                        -bf-cbc                   -bf-cfb                    -bf-ecb
-bf-ofb                    -blowfish                  -camellia-128-cbc         -camellia-128-cfb          -camellia-128-cfb1
-camellia-128-cfb8         -camellia-128-ecb          -camellia-128-ofb          -camellia-192-cbc          -camellia-192-cfb
-camellia-192-cfb1         -camellia-192-cfb8          -camellia-192-ecb         -camellia-192-ofb          -camellia-256-cbc
-camellia-256-cfb          -camellia-256-cfb1         -camellia-256-cfb8        -camellia-256-ecb          -camellia-256-ofb
-camellia128               -camellia192               -camellia256              -cast                      -cast-cbc
-cast5-cbc                 -cast5-cfb                 -cast5-ecb                -cast5-ofb                 -des
-des-cbc                   -des-cfb                   -des-cfb1                 -des-cfb8                  -des-ecb
-des-ede                   -des-ede-cbc               -des-ede-cfb              -des-ede-ofb               -des-ede3
-des-ede3-cbc              -des-ede3-cfb              -des-ede3-cfb1            -des-ede3-cfb8             -des-ede3-ofb
-des-ofb                   -des3                      -desx                     -desx-cbc                  -id-aes128-CCM
-id-aes128-GCM             -id-aes128-wrap            -id-aes192-CCM            -id-aes192-GCM             -id-aes192-wrap
-id-aes256-CCM             -id-aes256-GCM             -id-aes256-wrap           -id-smime-alg-CMS3DESwrap  -rc2
-rc2-40-cbc                -rc2-64-cbc                -rc2-cbc                  -rc2-cfb                   -rc2-ecb
-rc2-ofb                   -rc4                       -rc4-40                   -rc4-hmac-md5              -seed
-seed-cbc                  -seed-cfb                  -seed-ecb                 -seed-ofb
```

OpenSSL: шифрование алгоритмом AES-256

1. Сгенерируем ключ шифрования 32 байта (256 бит):

```
openssl rand -hex 32 > symmetric_key.hex
```

2. Подготовим текстовый файл (который будем шифровать):

```
nano sample.txt
```

3. Выполним команду шифрования:

```
openssl enc -aes-256-cbc -in sample.txt -out sample.enc \
        -pass file:symmetric_key.hex
```

Имя выходного (зашифрованного) файла sample.enc

OpenSSL: расшифрование алгоритмом AES-256

4. Проверим шифр-текст

```
cat sample.enc
```

5. Проведем расшифрование

```
openssl enc -d -aes-256-cbc -in sample.enc \
        -out sample.dec -pass file:symmetric_key.hex
```

Имя выходного (зашифрованного) файла sample.enc

количеству натуральных чисел, меньших либо равных и взаимно простых с ним

Асимметричное шифрование

1. Выбираем 2 простых числа: $p = 3$ $q = 11$

2. Вычисляем их произведение: $n = p \times q = 3 \times 11 = 33$ $n = 33$

3. Вычисляем функцию Эйлера: $\phi(n) = (p-1) \times (q-1) = 2 \times 10 = 20$ $\phi(n) = 20$

Количество натуральных чисел, меньших либо равных n и взаимно простых с ним

4. Выбираем «открытую» экспоненту: $1 < e < \phi(n)$, *и нет общих делителей с* $\phi(n)$ $e = 3$

Открытый ключ: $(e = 3, n = 33)$

5. Вычисляем «закрытую» экспоненту: $d \times e \bmod \phi(n) = 1$ $(d \times 3) \bmod 20 = 1$ $d = 7$

Закрытый ключ: $(d = 7, n = 33)$

Асимметричное шифрование

Открытый ключ:

$(e = 3, n = 33)$

Закрытый ключ:

$(d = 7, n = 33)$

Формула шифрования:

$$C = M^e \ mod \ n$$

Формула расшифрования:

$$M = C^d \ mod \ n$$

Текст (M): 4

Шифруем:

$$C = 4^3 \ mod \ 33 = 64 \ mod \ 33 = 31$$

Шифр-текст (C): 31

Шифр-текст (C): 31

Расшифруем:

$$\mathrm{M} = 31^7 \ mod \ 33 = 27512614111 \ mod \ 33 = 4$$

Текст (M): 4

Асимметричное шифрование

Как решение задачи обмена ключами симметричного шифрования, особенно в компьютерных сетях

Закрытый и открытый ключ – простые числа.

Закрытый ключ (private) ✕ Открытый ключ (public)

Алгоритмы: RSA, DSA, ECDSA

\+ Позволяет обмениваться ключами шифрования по открытым каналам (небезопасным)

\− Медленно, из-за этого не используется для шифрования текстов (файлов), а только для установления соединений

Использование в Secure протоколах (HTTPS, SMTPS,...)

Установка соединения TCP

Получение сертификата (с открытым ключом сервера) → Проверка сертификата по подписи центра сертификации (или самоподписанный)

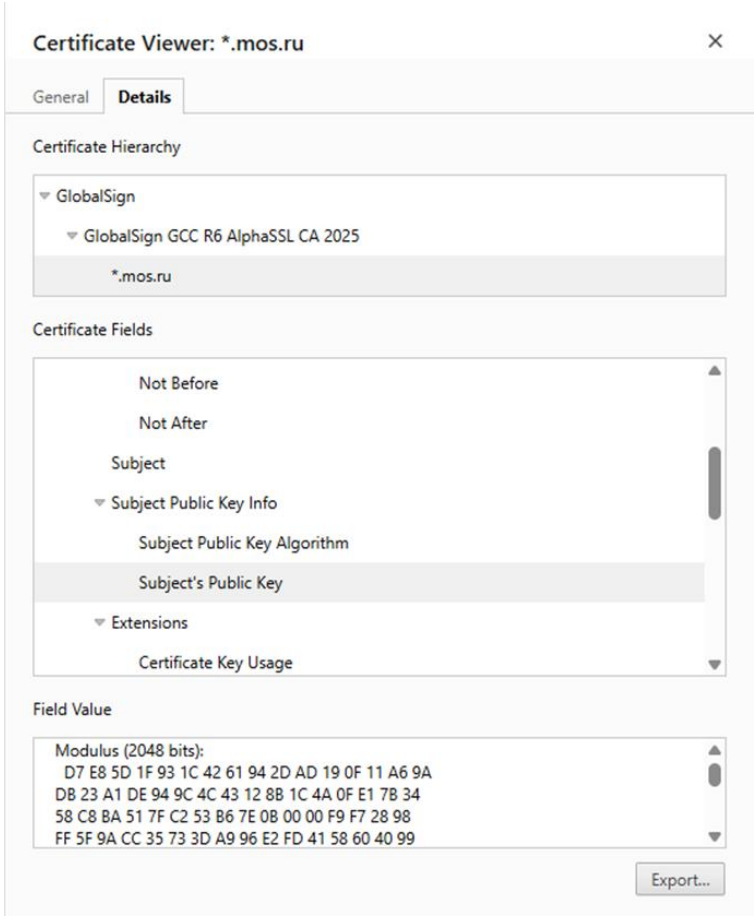
Выработка сеансового ключа шифрования (обмен ключами)

Симметричное шифрование данных в потоке (вот тут основной протокол)

Использование и создание сертификатов в openSSL

«Сертификат» - набор данных, включая публичный ключ, с помощью которого можно устанавливать безопасные соединения, а также дополнительная информация:

- издатель (issuer),
- срок действия ,
- цифровая подпись,
- центр сертификации, выдавший его (CA)



Использование и созд

Проверить сертификат на любом сервере и л

```
openssl s_client -connect mos.r
```

```
---
Certificate chain
 0 s:/CN=*.mos.ru
   i:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign GCC R6 AlphaSSL CA 2025
 1 s:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign GCC R6 AlphaSSL CA 2025
   i:/OU=GlobalSign Root CA - R6/O=GlobalSign/CN=GlobalSign
 2 s:/OU=GlobalSign Root CA - R6/O=GlobalSign/CN=GlobalSign
   i:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
 3 s:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
   i:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIGSjCCBTKgAwIBAgIMGNqFbrH+qZKfUzd1MA0GCSqGSIb3DQEBCwUAMFUxCzAJ
BgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMSswKQYDVQQDEyJH
bG9iYWxTaWduIEdDQyBSNiBBbHBoYVNTTCBDQSAyMDI1MB4XDTI1MTAxNDEzMzg0
M1oXDTI2MTExNTEzMzg0MlowEzERMA8GA1UEAwwIKi5tb3MucnUwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDX6F0fkxxCYZQtrRkPEaaa2yOh3pScTEMS
ixxKD+F7NFjIulF/wlO2fgsAAPn3KJj/X5rMNXM9qZbi/UFYYECZjNAPFzh/nMQM
6oy1MUkaMIp0NNyWwEAaOhJTHC5ZWS8TT/I/X2b5+nrMfdNLkIgENJFBaNyxkBMV
dj9dOBPrEVB6mJus/HgzyuFyu9mLLR5vpiGEUcDqlNtOh75FsOo6OhjcHG5H1QC7
BO9otjR8XGdoggDQAAOEFzeFrdZSjOxwG1SNKE36oeh9IQQoO/UApNjJLAoOJvpN
j2MuBm/YOu6EejudOinLCHyLFOoqVvX6no+BVI4hdunUFuMlCM5XAgMBAAGjggNa
MIIDVjAOBgNVHQ8BAf8EBAMCBaAwDAYDVR0TAQH/BAIwADCBmQYIKwYBBQUHAQEE
gYwwgYkwSQYIKwYBBQUHMAKGPWh0dHA6Ly9zZWN1cmUuZ2xvYmFsc2lnbi5jb20v
Y2FjZXJ0L2dzZ2NjcjZhbHBoYXNzbGNhMjAyNS5jcnQwPAYIKwYBBQUHMAGGMGh0
dHA6Ly9vY3NwLmdsb2JhbHNpZ24uY29tL2dzZ2NjcjZhbHBoYXNzbGNhMjAyNTBX
BgNVHSAEUDBOMAgGBmeBDAECATBCBgorBgEEAaAyCgEDMDQwMgYIKwYBBQUHAgEW
Jmh0dHBzOi8vd3d3Lmdsb2JhbHNpZ24uY29tL3JlcG9zaXRvcnkvMEQGA1UdHwQ9
MDswOaA3oDWGM2h0dHA6Ly9jcmwuZ2xvYmFsc2lnbi5jb20vZ3NnY2NyNmFscGhh
c3NsY2EyMDI1LmNybDAbBgNVHREEFDASggggLm1vcy5ydYIGbW9zLnJ1MB0GA1Ud
JQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAfBgNVHSMEGDAWgBTFtJOPbyvcHki/
txAwhc7RsrtILTAdBgNVHQ4EFgQU6gSz27J/OB2Pdt9FKjGf5d4YhIOwggF9Bgor
BgEEAdZ5AgQCBIIBbQSCAWkBZwB2AJROQ4f67MHvgfMZJCaoGGUBx9NfOAIBP3Jn
fVU3LhnYAAABmeLx4VQAAAQDAEcwRQIhAMit3xtTkQpGc3bpRigNcK5+lvIIDHzh
EX7Oa/xWkUcvAiAxna+VMJ2VXE+E2ozjNrWiXztUQsW9kjp8/lP+j9XjGQB1AKyr
MHBs6+yEMfQTOvSRXxEeQiRDsfKmjE88KzunHgLDAAABmeLx4VAAAAQDAEYwRAIg
Iu6dELi4taX/PtP4K8WatLFUpDi6R62320ZXQCxwgqECIGMlcIsKryuZPv6X5ck6
QMfMKqI4/+1MQLh9L+T5U1Y7AHYAyzj3FYl8hKFEX1vB3fvJbvKaWc1HCmkFhbDL
FMMUWOcAAAGZ4vHhbgAABAMARzBFAiAV4UPw+HGBPihd8hH89mJdp+WZL6F/neJY
YZS8gR3jqAIhAIQ9iGrUCucuSfSo4tujNbvDvwHEfDMszIIHwH5zS6NTMA0GCSqG
SIb3DQEBCwUAA4IBAQALNAfIcf73bZKqO1CtGZtPIq8aYEUIldo8x3GcaIoSIa62
TZ2KvID6CILy1CaqO6wnjAWvNRkEePDgtpx9WlzOIFGIpqlh/jOOQpbkqilDgfSC
rXi5Nn8rBmv8dnFby/V6KVayhT2YOfN6/xGdywLN2NDRWmGzRVf5wxu2+730veJ2
tdqRYGdi3GTHvJRPOhXSBSAOdV/Yy0Uv409WDdum6xrbkabMqTcHAdf6h7am6BVI
dBqEiD1MFtpQ3aZnLKIzUL5fn1+8Uk/rYtcVclTJVrhHNnhq2zEFxSy4AmKHpzIZ
yVO2xbaCHvfymV/P3tIqHwhBOgyvfz6oHkaT9IYw
-----END CERTIFICATE-----
subject=/CN=*.mos.ru
issuer=/C=BE/O=GlobalSign nv-sa/CN=GlobalSign GCC R6 AlphaSSL CA 2025
---
No client certificate CA names sent
Peer signing digest: SHA256
Server Temp Key: ECDH, P-256, 256 bits
---
SSL handshake has read 5972 bytes and written 391 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID: 3E223B5546F3A88B345E134C9CB2CE16A4F0D1B9891F5F6272711CB1772F39D8
    Session-ID-ctx:
    Master-Key: DF3874FBA87A88265F728A7384540E7E433043637294D28F021AA7126D875520E08E67AF1C9DD6A39BE05A76B03FAF82
    Key-Arg   : None
```

Создание сертификата

1. Подготовим приватный ключ:

```
openssl genrsa -out private.key 2048
```

2. Выполним формирование запроса на сертификат:

```
openssl req -new -key private.key -out request.csr
```

```
root@ajkelf-cloud-vm2:/home/joseph/tt# openssl req -new -key private.key -out request.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:MOW
Locality Name (eg, city) []:Moscow
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MOS
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:TEST
Email Address []:test@test.ru

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@ajkelf-cloud-vm2:/home/joseph/tt#
```

Создан

3. Создание самоподписанного сертификата:

```
openssl x509 -req -days 365 -in
      -signkey private.key -ou
```

4. Проверка содержимого сертификата:

```
openssl x509 -in certificate.c
```

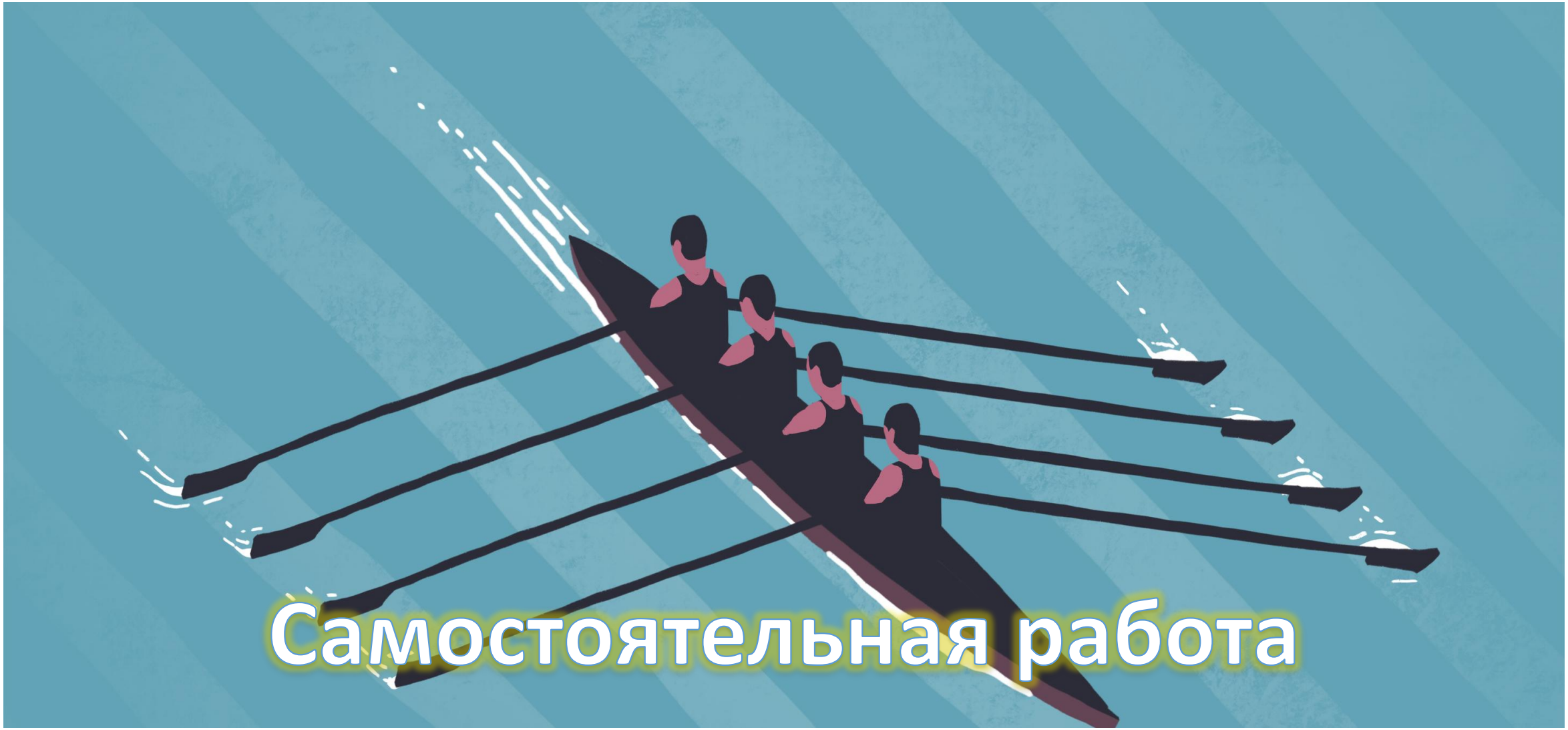
```
root@ajkelf-cloud-vm2:/home/joseph/tt# openssl x509 -in certificate.crt -text -noout
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 10029467403508533700 (0x8b2fd377107c49c4)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=RU, ST=MOW, L=Moscow, O=MOS, CN=TEST/emailAddress=test@test.ru
        Validity
            Not Before: Jan  2 22:10:04 2026 GMT
            Not After : Jan  2 22:10:04 2027 GMT
        Subject: C=RU, ST=MOW, L=Moscow, O=MOS, CN=TEST/emailAddress=test@test.ru
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d2:b7:15:d6:bc:55:c0:c1:4c:95:84:70:18:fb:
                    3c:ab:1e:39:e2:ee:00:25:f3:bb:75:8c:29:5f:28:
                    45:cf:27:09:6c:49:c3:55:0e:a6:c3:79:b1:7c:44:
                    67:39:78:47:25:ab:b3:f0:4b:a2:a0:78:ba:e4:50:
                    05:44:c4:23:83:fd:40:31:6f:5e:25:a8:0a:e8:3f:
                    3c:d2:3b:aa:5f:10:03:54:63:9d:be:50:b6:1b:37:
                    3d:a3:6f:7f:d3:be:0e:a2:55:cc:5e:59:4f:ee:79:
                    33:b5:68:05:28:0e:80:9e:1b:59:ea:c6:fb:83:9e:
                    d1:3d:d7:5d:50:e6:2d:bb:33:53:7e:c1:5d:6d:1a:
                    e3:e6:9e:95:47:99:d5:90:4d:eb:1a:dc:2a:f5:9d:
                    53:1f:2b:5f:52:bd:c6:c9:a9:f3:21:b9:23:09:d6:
                    57:2d:82:6c:1d:6d:f0:55:82:50:63:fd:2d:55:04:
                    1f:a5:12:2a:5e:36:99:b9:d2:19:96:ae:c3:b6:f7:
                    a8:c1:92:45:86:24:47:a3:3a:9a:2b:ea:8a:ce:a7:
                    67:31:5b:3f:f5:04:11:ef:8f:11:9c:c6:a3:ce:0d:
                    48:f0:b6:bc:b5:e8:d5:5d:e5:f9:b8:ab:5f:26:89:
                    4e:72:9e:55:a9:d2:bc:ce:7c:63:fa:68:4c:93:f6:
                    91:63
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
         8a:04:83:e5:b7:50:51:fc:6e:aa:e1:c8:ac:6b:b8:c9:76:81:
         a8:15:b8:6c:d7:0e:09:c2:24:04:ac:fc:43:b4:d7:f6:d3:fd:
         2a:44:80:0f:38:ed:7d:61:f4:0a:43:89:80:8c:3e:5f:b2:be:
         07:fa:b4:f0:24:5c:33:19:1c:29:09:a1:54:9e:eb:9d:5f:a0:
         ec:3d:57:45:a8:e3:fc:66:8f:e1:aa:ec:15:c9:15:19:36:4e:
         50:2c:a2:58:69:29:9c:25:f6:aa:1f:9e:92:9a:8c:44:5b:77:
         49:6c:9e:0a:95:0d:89:6e:97:5e:52:85:b2:48:41:03:b3:7b:
         a8:6e:f5:62:8b:f8:95:c3:e3:46:c5:6d:92:fd:56:99:97:ce:
         23:49:d1:be:70:fc:c3:66:74:09:f1:6c:6e:19:91:ab:b6:7d:
         17:3b:c0:a7:13:dd:fa:db:9a:b7:86:ac:22:4a:17:b7:b0:d0:
         db:43:5d:a6:7d:7d:6a:79:72:21:cc:a3:4c:35:38:88:40:87:
         b9:7d:5f:cf:61:79:ff:a6:99:58:45:41:92:a4:a1:48:25:64:
         85:9e:9b:fb:80:7b:e9:f5:7e:a3:d8:38:5f:56:1e:e6:2b:9f:
         aa:7c:9e:c1:84:f2:c3:be:f6:30:11:bb:71:5f:b5:59:47:e0:
         87:b2:e6:ae
root@ajkelf-cloud-vm2:/home/joseph/tt#
```

Применение ассиметричного шифрования

Безопасный обмен ключами (симметричного) шифрования → Протоколы: HTTPS, SMTPS, IMAPS, LDAPS, POP3S, SSH, …

Цифровая подпись (производится шифрование с помощью закрытого ключа, можно проверить авторство с помощью открытого ключа)

Блокчейн (и криптовалюты)


Самостоятельная работа

Самостоятельная работа по рискам ИБ

1. Тему рисков ИБ мы рассмотрим на следующей лекции

2. Каждый студент состоит в команде (в которой сдавал предыдущую контрольную по инцидентам)

3. Каждой команде будет выделена организация (реальная), по которой нужно будет провести изучение ИТ-инфраструктуры

Распределим организации на семинарах

4. Внутри команды, каждый студент выбирает для самостоятельной работы уникальную информационную систему в организации.

5. Внутри команды, каждый студент выбирает для самостоятельной работы уникальную информационную систему в организации. По ней будет задание на оценку рисков ИБ.

Что будет на семинаре № 5?

- Обсуждение по прошедшим темам лекций
- Обсуждение результатов тестов
- Подготовка к самостоятельной работе по оценке рисков
- Тест по теме «шифрование»


ВЫСШАЯ • ШКОЛА • ЭКОНОМИКИ •