Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе по теме: “Криптография”

по дисциплине «**Информационная безопасность**»

Авторы: Юрпалов С. Н.
Кошкин М. С.

Факультет: ИТиП

Группа: М34051

Санкт-Петербург 2023

Ход работы:

0) Настройка

Сергей

```
wilfordaf@Ubuntu-22:~$ gpg --version
gpg (GnuPG) 2.2.27
libgcrypt 1.9.4
Copyright (C) 2021 Free Software Foundation, Inc.
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/wilfordaf/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

Михаил

```
(base) root@lorelie:/mnt/d# gpg --version
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg (GnuPG) 2.2.27
libgcrypt 1.9.4
Copyright (C) 2021 Free Software Foundation, Inc.
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

1) Создание ключевой пары

Настройте gpg на работу с сервером ключей hkps://keyserver.ubuntu.com

Добавим в *путь_к_директории_gnump*/gpg.conf

keyserver hkps://keyserver.ubuntu.com

Михаил

```
(base) root@lorelie:~# mkdir .gnupg
(base) root@lorelie:~# nano .gnupg/gpg.conf
(base) root@lorelie:~# cat .gnupg/gpg.conf
keyserver hkps://keyserver.ubuntu.com
```

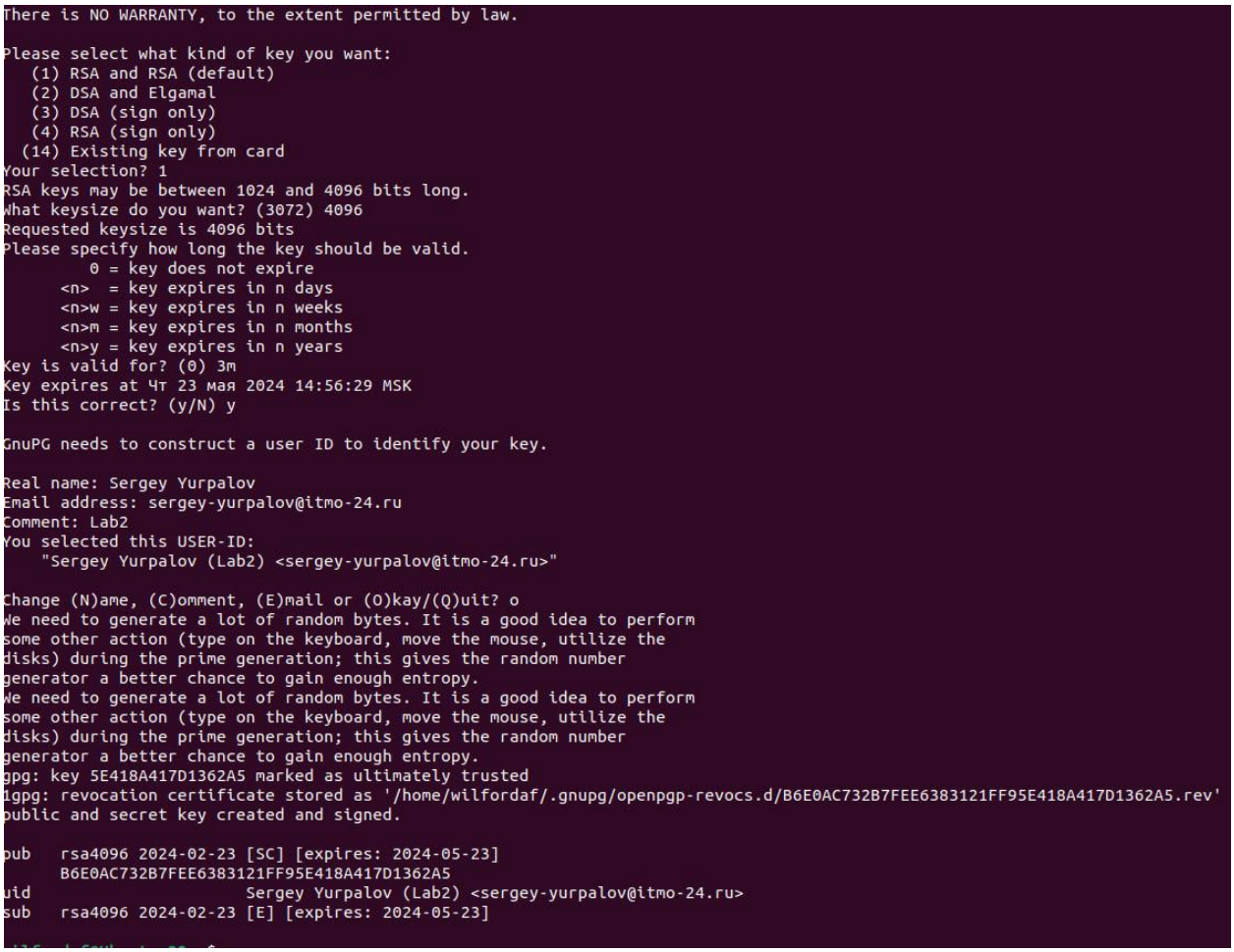
Сергей

```
wilfordaf@Ubuntu-22:~$ cat /home/wilfordaf/.gnupg/gpg.conf
keyserver hkps://keyserver.ubuntu.com
wilfordaf@Ubuntu-22:~$
```

Создайте пару ключей RSA, которыми можно подписывать и шифровать файлы. Длину ключа укажите 4096 бит. Время жизни 3 месяца.

Используем команду gpg --full-gen-key

Сергей



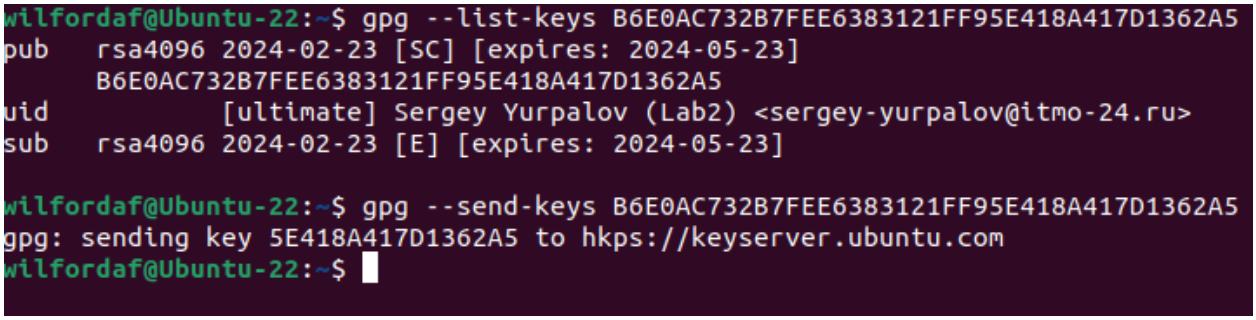
Михаил

```
(base) root@lorelie:~# gpg --full-gen-key
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 3m
Key expires at Thu May 23 15:01:52 2024 MSK
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Mikhail Koshkin
Email address: mikhail-koshkin@itmo-24.ru
Comment: Lab2
You selected this USER-ID:
    "Mikhail Koshkin (Lab2) <mikhail-koshkin@itmo-24.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
12We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
dfdfdmfgdkfkdgndngpg: key 0FD137640E9CC76F marked as ultimately trusted
jhbdbfgpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/D059C9845691FD7C348C462E0FD137640E9CC76F.rev'
public and secret key created and signed.

pub   rsa4096 2024-02-23 [SC] [expires: 2024-05-23]
      D059C9845691FD7C348C462E0FD137640E9CC76F
uid                      Mikhail Koshkin (Lab2) <mikhail-koshkin@itmo-24.ru>
sub   rsa4096 2024-02-23 [E] [expires: 2024-05-23]
```

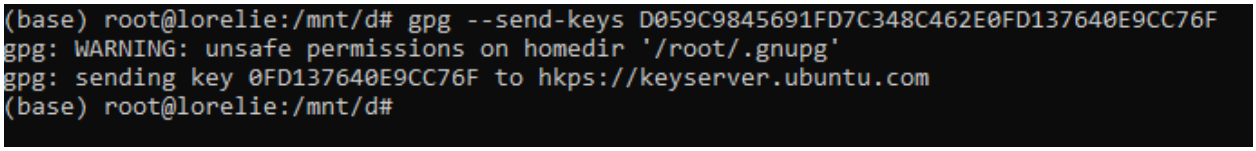
Опубликуйте свой ключ на сервере ключей.

Используем команду: gpg --send-keys <key-id>

Сергей



Михаил



Запомните fingerprint своего публичного ключа.

Используем команду: gpg --fingerprint <key-id>

Сергей

```
wilfordaf@Ubuntu-22:~$ gpg --fingerprint B6E0AC732B7FEE6383121FF95E418A417D1362A5
pub   rsa4096 2024-02-23 [SC] [expires: 2024-05-23]
      B6E0 AC73 2B7F EE63 8312  1FF9 5E41 8A41 7D13 62A5
uid           [ultimate] Sergey Yurpalov (Lab2) <sergey-yurpalov@itmo-24.ru>
sub   rsa4096 2024-02-23 [E] [expires: 2024-05-23]

wilfordaf@Ubuntu-22:~$ nano Documents/fingerprint.txt
wilfordaf@Ubuntu-22:~$ cat Documents/fingerprint.txt
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2024-05-23
pub   rsa4096 2024-02-23 [SC] [expires: 2024-05-23]
      B6E0AC732B7FEE6383121FF95E418A417D1362A5
uid           [ultimate] Sergey Yurpalov (Lab2) <sergey-yurpalov@itmo-24.ru>
sub   rsa4096 2024-02-23 [E] [expires: 2024-05-23]
```

Михаил

```
(base) root@lorelie:/mnt/d# gpg --fingerprint D059C9845691FD7C348C462E0FD137640E9CC76F
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
pub   rsa4096 2024-02-23 [SC] [expires: 2024-05-23]
      D059 C984 5691 FD7C 348C  462E 0FD1 3764 0E9C C76F
uid           [ultimate] Mikhail Koshkin (Lab2) <mikhail-koshkin@itmo-24.ru>
sub   rsa4096 2024-02-23 [E] [expires: 2024-05-23]

(base) root@lorelie:/mnt/d# cd ~
(base) root@lorelie:~# nano fingerprint.txt
(base) root@lorelie:~# cat fingerprint.txt
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2024-05-23
pub   rsa4096 2024-02-23 [SC] [expires: 2024-05-23]
      D059C9845691FD7C348C462E0FD137640E9CC76F
uid           [ultimate] Mikhail Koshkin (Lab2) <mikhail-koshkin@itmo-24.ru>
sub   rsa4096 2024-02-23 [E] [expires: 2024-05-23]
(base) root@lorelie:~#
```

2) Шифрованный обмен

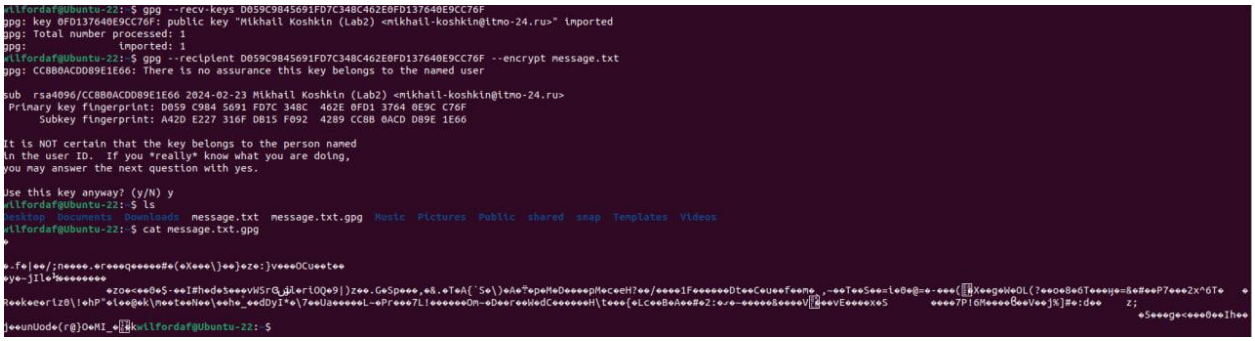
Сергей → Михаил

Создаём сообщение: Hello world!

```
wilfordaf@Ubuntu-22:~$ nano message.txt
wilfordaf@Ubuntu-22:~$ cat message.txt
Hello world!
```

Загружаем ключ получателя – Михаила: gpg --recv-keys <recipient-key-id>

Шифруем файл: gpg --recipient <recipient-key-id> --encrypt <filename>



Передаём зашифрованный файл Михаилу.

Расшифровываем файл: gpg –decrypt <filename>

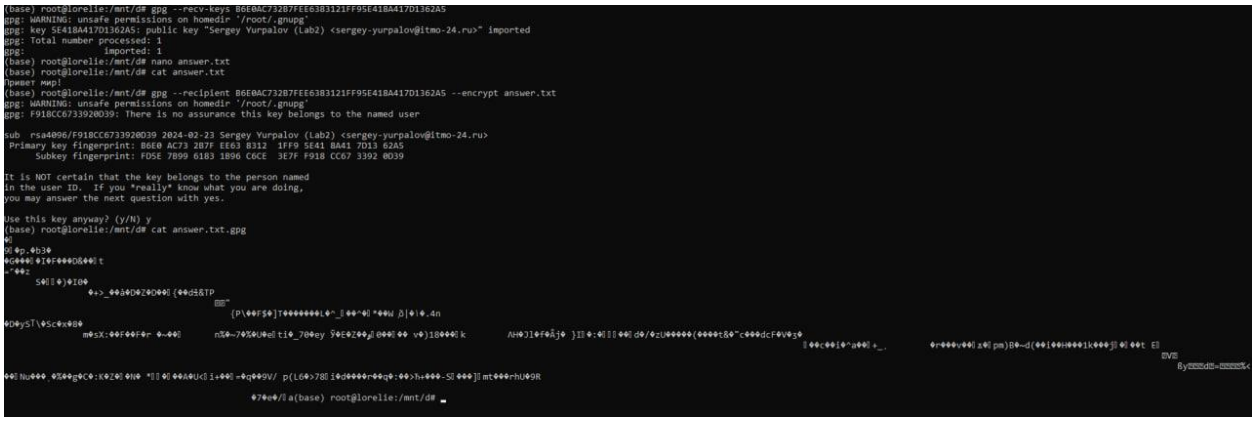
```
(base) root@lorelie:/mnt/d# gpg --decrypt message.txt.gpg
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: encrypted with 4096-bit RSA key, ID CC8B0ACDD89E1E66, created 2024-02-23
      "Mikhail Koshkin (Lab2) <mikhail-koshkin@itmo-24.ru>"
Hello world!
(base) root@lorelie:/mnt/d#
```

Сергей ← Михаил

Создаём сообщение: Привет мир!

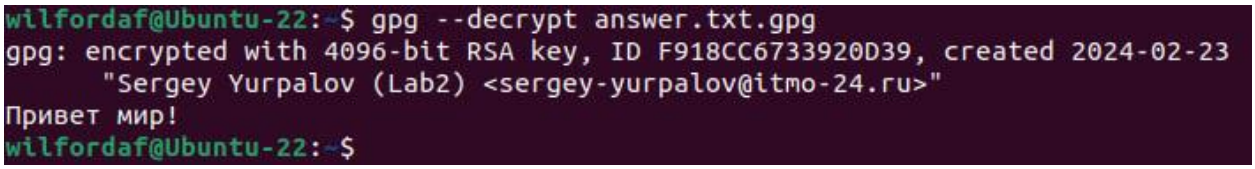
Загружаем ключ получателя – Сергея: gpg --recv-keys <recipient-key-id>

Шифруем файл: gpg --recipient <recipient-key-id> --encrypt <filename>



Передаём зашифрованный файл Сергею.

Расшифровываем файл: gpg –decrypt <filename>

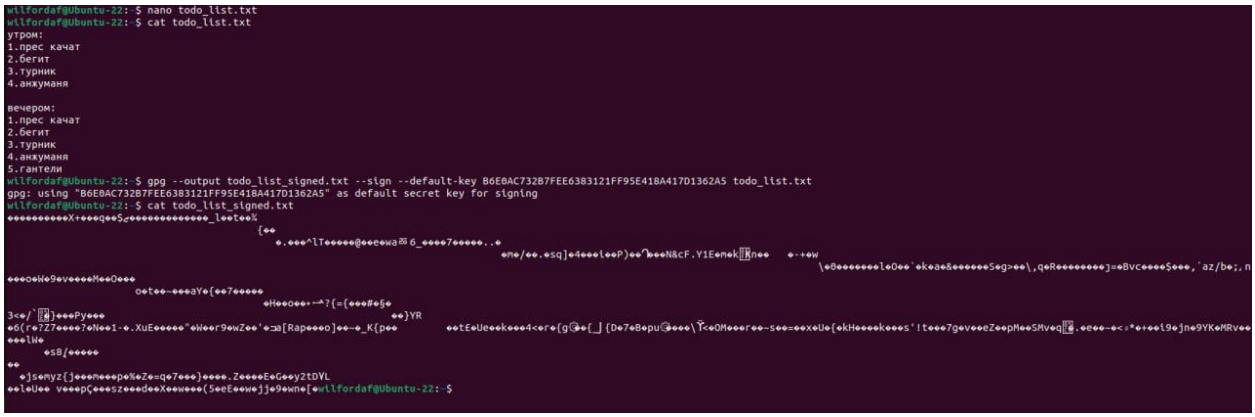


3) Электронная подпись

Сергей → Михаил

Создадим файл todo_list.txt с расписанием.

Создадим файл подписи: gpg --output todo_list.sig --detach-sig todo_list.txt



Передадим оба файла Михаилу.

Проверим подпись: gpg --verify todo_list.sig todo_list.txt

```
(base) root@lorelie:/mnt/d# gpg --verify todo_list.sig todo_list.txt
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: Signature made Fri Feb 23 16:24:15 2024 MSK
gpg:                using RSA key B6E0AC732B7FEE6383121FF95E418A417D1362A5
gpg: Good signature from "Sergey Yurpalov (Lab2) <sergey-yurpalov@itmo-24.ru>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: B6E0 AC73 2B7F EE63 8312  1FF9 5E41 8A41 7D13 62A5
(base) root@lorelie:/mnt/d#
```

Изменим файл и проверим подпись снова:

```
(base) root@lorelie:/mnt/d# nano todo_list.txt
(base) root@lorelie:/mnt/d# cat todo_list.txt
вечером:
1.прес качат
2.бегит
3.турник
4.анжуманя

утром:
1.прес качат
2.бегит
3.турник
4.анжуманя
5.гантели
(base) root@lorelie:/mnt/d# gpg --verify todo_list.sig todo_list.txt
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: Signature made Fri Feb 23 16:24:15 2024 MSK
gpg:                using RSA key B6E0AC732B7FEE6383121FF95E418A417D1362A5
gpg: BAD signature from "Sergey Yurpalov (Lab2) <sergey-yurpalov@itmo-24.ru>" [unknown]
(base) root@lorelie:/mnt/d#
```

Ответы на вопросы:

1. Для шифрования файлов обычно используется открытый ключ получателя. Это гарантирует, что расшифровать и получить доступ к содержимому зашифрованного файла сможет только тот получатель, который владеет соответствующим закрытым ключом. Поэтому при шифровании файлов следует использовать открытый ключ того человека или организации, с которыми вы хотите безопасно поделиться информацией.
2. Для подписи файлов вы используете свой собственный закрытый ключ. Подписывая файл своим личным ключом, вы создаете цифровую подпись, которая может быть проверена с помощью вашего открытого ключа. Это позволяет другим подтвердить, что файл действительно был подписан вами и что он не был изменен с момента подписания. Поэтому при подписании файлов следует использовать собственный закрытый ключ для проверки подлинности и обеспечения целостности содержимого.
3. Можно использовать такой алгоритм:
    - Вычислим криптографический хэш (например, SHA-256) содержимого документа. Этот хэш служит уникальным идентификатором документа.
    - Каждая сторона генерирует свою собственную цифровую подпись для хэша документа, используя свой закрытый ключ. (подпись создается путем шифрования хэша с помощью закрытого ключа стороны)
    - Добавим цифровую подпись каждой стороны к документу вместе с ее идентификационной информацией (например, именем, электронной почтой).
    - Распространим документ среди всех участвующих сторон, чтобы каждая сторона получила одну и ту же копию документа с прикрепленным блоком подписи.
    - Каждая сторона расшифрует каждую подпись с помощью открытого ключа подписавшей стороны, чтобы получить исходный хэш документа.
    - Если хэши совпадают, то подпись действительна, что свидетельствует о том, что документ не был изменен с момента его подписания.