12345 Алексей

КриптоАРМ API

2020 г

Оглавление

Описание API КриптоАРМ

Доступно множество команд, которые взаимодействуют с КриптоАРМ. Все они открывают КриптоАРМ, если он не запущен. Их можно ввести через адресную строку браузера (вы можете размещать их так же, как ссылки на веб-страницы) или в терминале (для Windows интерпретатор команд). Для взаимодействия используется зарегистрированный протокол **cryptoarm://**

В текущей редакции доступны команды:

- signAndEncrypt – выполнение криптографических операций над документами (подпись, шифрование, проверка подписи, расшифрование)
- certificates – экспорт или импорт сертификатов, просмотр свойств сертификата
- certrequests – генерация запросов на сертификат
- diagnostics – диагностика рабочего места
- startView – открыть окно или вкладку

Общий сценарий выполнения команд (для взаимодействия с web-приложениями):

1. Пользователь заходит на портал (web-приложение).
2. Выбирает объекты (например список документов) и действие (например подпись).
3. Портал генерирует и отображает (или сразу переходит) ссылку с протоколом cryptoarm://
4. Если КриптоАРМ не запущен, то запускается. Затем обращается к порталу за JSON с набором параметров, нужных для выполнения конкретной операции. JSON генерируется на сервере, где располагается web-приложение.
5. Полученный JSON обрабатывается и в зависимости от команды выполняются нужные дополнительные запросы к web-приложению.
6. Пользователь подтверждает саму запрошенную операцию (остальной функционал приложения блокируется).
7. Результаты отправляются на сервер.

Общий формат ссылки:

cryptoarm://<command>/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **<command>** - выполняемая команда
- **<URL>** - ссылка на получение JSON с параметрами, нужными для выполнения команды
- **?id=<id>** - обязательный параметр. Идентификатор транзакции.

1. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. ДОЛЖЕН использоваться TLS, незащищенные соединения КриптоАРМ отклоняет.

POST запрос

КриптоАРМ выполняет HTTP POST запросы, которые содержат заголовки:

- Content-Type: ДОЛЖЕН быть application/json.
- Content-Length: ДОЛЖЕН содержать правильную длину в соответствии с HTTP-спецификацией.
- Accept: ДОЛЖЕН быть application/json.

GET запрос

Не используются.

Ответ

HTTP ответ сервера ДОЛЖЕН содержать заголовки:

- Content-Type: ДОЛЖЕН быть application/json.
- Content-Length: ДОЛЖЕН содержать правильную длину в соответствии с HTTP-спецификацией.

Объект Error

В случае ошибки сервер ДОЛЖЕН отправить ответ следующей структуры:

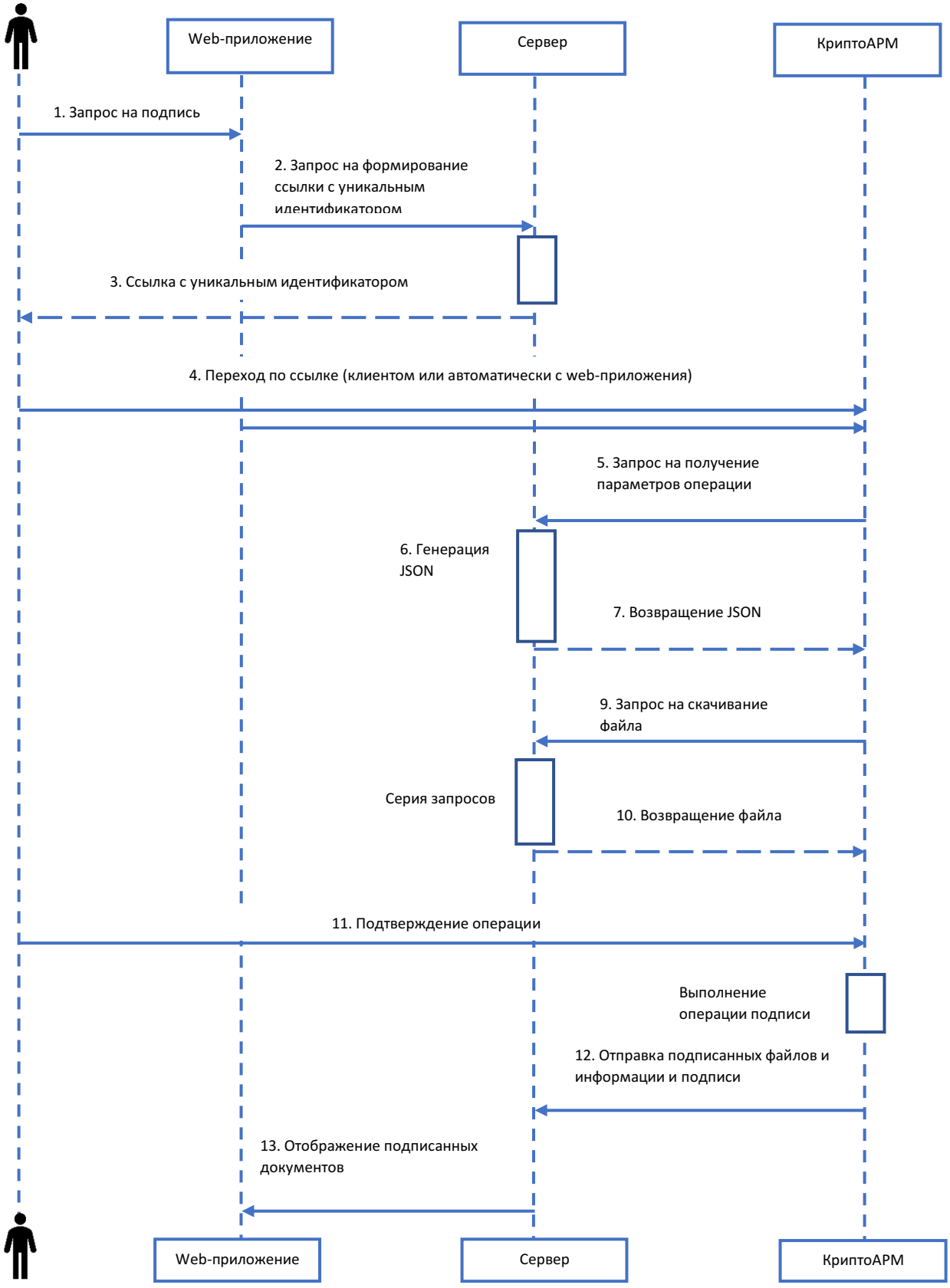
Ключ	Тип	Описание
code	number	Код ошибки
message	string	Короткое описание ошибки
data	string | Object	Необязательное поле. Может содержать дополнительные сведения об ошибке

HTTP-коды

Код	Ошибка	Описание
200	OK	И для ответов, и для ошибок
204	No Response	Для пустых запросов (нотификация)
405	Method Not Allowed	Метод не доступен
415	Unsupported Media Type	Если Content-Type не application/json

2. Команда signAndEncrypt. Запросы на подпись и шифрование документов, выполнение обратных операций

Команда **signAndEncrypt** (подпись и шифрование) используется для запроса на подпись документа или пакета документов. Может использоваться в качестве аутентификатора. Выполнение операции требует действующей лицензии на КриптоАРМ ГОСТ. Схема:



2.1. Формат ссылки

Для выполнения команды certificates должна быть сформирована ссылка вида:

cryptoarm://signAndEncrypt/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **signAndEncrypt** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://signAndEncrypt/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

2.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе 1. Описание запросов и ответов.

2.2.1. Получение параметров операции

После получения команды **signAndEncrypt** КриптоАРМ отправляет запрос на <URL> для получения параметров операции.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.parameters»	Используемый метод. Всегда «signAndEncrypt.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
diagnostic	IDiagnosticInformaton	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "signAndEncrypt.parameters",
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
```

```
      }
}
```

Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	ISignAndEncryptParameters	Объект со сведениями о параметрах операции подписи
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)

Пример ответа:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": [
            "SIGN",
            "ARCHIVE",
            "ENCRYPT"
        ],
        "props": {
            "headerText": "Подпись документов cryptoarm.ru",
            "license": "",
            "files": [{
                    "name": "file1.txt",
                    "url": "http://localhost:8080/public/files/file1.txt",
                    "id": 1,
                    "urlDetached": ""
                },
                {
                    "name": "file2.txt",
                    "url": "http://localhost:8080/public/files/file2.txt",
                    "id": 2,
                    "urlDetached": ""
                },
                {
                    "name": "file4.pdf",
                    "url": "http://localhost:8080/public/files/file4.pdf",
```

```
                    "id": 4,
                    "urlDetached": ""
                }
            ],
            "extra": {
                "token": "9c7101f7-9c47-4481-b4da-a6a497abde08",
                "signType": "1",
                "signStandart": "1"
            },
            "uploader": "http://localhost:8080/upload"
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

2.2.2. Отправка результата прямых операций

После того, как пользователь выберет нужные сертификаты КриптАРМ выполняет операцию. Полученные файлы отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.outDirectResults»	Используемый метод. Всегда «signAndEncrypt.outDirectResults»
params	Объект типа IDirectResults	Сведения о результатах прямой операции

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "signAndEncrypt.outDirectResults",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "directResults": [
          {
            "out": "MIIWgQYJKoZIhvcNAQcCoIIWcjCCFm4CA…cN/aHmA="
          }
        ]

    }
```

```
}
```

2.2.3. Отправка результата обратных операций

Результаты отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.outReverseResults»	Используемый метод. Всегда «signAndEncrypt.outReverseResults»
params	Объект типа IReverseResults	Сведения о результатах обратной операции

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "signAndEncrypt.outReverseResults",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "reverseResults": [
          {
            "out": "MIIWgQYJKoZIhvcNAQcCoIIWcjCCFm4CA…cN/aHmA="
          }
        ]

    }
}
```

2.2.4. Отправка результата проверки подписи

Результаты проверки подписи отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»

method	«signAndEncrypt.verifySignResults»	Используемый метод. Всегда «signAndEncrypt.verifySignResults»
params	Объект типа IVerifySignResults	Сведения о проверке подписи

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "signAndEncrypt.verifySignResults",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "verifyResults": [
          {
            "id": 2,
            "status": false,
            "signers": [
              {
                "hash": "MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV",
                "issuerFriendlyName": "Минкомсвязь России",
                "issuerName": "Минкомсвязь России",
                "subjectFriendlyName ": "Минкомсвязь России",
                "subjectName": "Минкомсвязь России",
                "status": true
              }
             ]
          }
        ]

    }
}
```

2.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **signAndEncrypt.**

2.3.1. Интерфейс ISignAndEncryptParameters

Объекты данного типа описывают вид операции и её параметры

Свойство	Тип	Описание
operation	string[]	Тип операции. Доступные значения типов (комбинировать различные типы нельзя): ISignAndEncryptOperationDirect, ISignAndEncryptOperationReverse,

		ISignAndEncryptOperationVerify
props	ISignAndEncryptOperationProps	Параметры операции

2.3.2. Тип ISignAndEncryptOperationDirect

Возможные прямые операции.

Значение	**Описание**
SIGN	Подпись
ARCHIVE	Архивирование
ENCRYPT	Шифрование

2.3.3. Тип ISignAndEncryptOperationReverse

Возможные обратные операции.

Значение	**Описание**
UNSIGN	Снятие подписи
DECRYPT	Расшифрование
UNZIP	Разархивирование

2.3.4. Тип ISignAndEncryptOperationVerify

Проверка подписи.

Значение	**Описание**
VERIFYSIGN	Проверка подписи

2.3.5. Интерфейс ISignAndEncryptOperationProps

Интерфейс ISignAndEncryptOperationProps описывает параметры операции.

Свойство	**Тип**	**Описание**
headerText?	string	Необязательные параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательные параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов
license?	string	Необязательное свойство. Содержит временную лицензию, которая будет использоваться для выполнения операции в КриптоАРМ
uploader	string	Ссылка, на которую будут отправлены результаты операции
files	Массив типа IFile[]	Массив файлов на подпись
extra	Объект типа IExtra	Настройки операции

2.3.6. Интерфейс IFile

Интерфейс IFile описывает файлы и ссылки на них.

Свойство	Тип	Описание
name	string	Имя файла (с расширением)
url	string	Ссылка на скачивание файла
id	string	Уникальный идентификатор файла
urlDetached?	string	Необязательный параметр. Используется для откреплённой подписи

2.3.7. Интерфейс IExtra

Интерфейс IExtra описывает настройки операции. Если параметр не задан, то пользователю доступен выбор из всех доступных в приложении значений.

Свойство	Тип	Описание
signType	number	Необязательный параметр. Возможные значения: 0 - присоединенная подпись 1 - отсоединённая подпись
signStandart	number	Необязательный параметр. Стандарт подписи. Возможные значения: 0 - CMS 1 - CaDES-X Long Type1
signEncoding	number	Необязательный параметр. Кодировка. Возможные значения: 0 – BASE-64 1 - DER
timestampOnSign	string	Необязательный параметр. Штамп времени на подпись. Возможные значения: True - будет добавлен штамп времени False – не будет добавлен штамп времени
timestampOnData	string	Необязательный параметр. Штамп времени на подписываема данные. Возможные значения: True - будет добавлен штамп времени False – не будет добавлен штамп времени
encryptEncoding	number	Необязательный параметр. Кодировка. Возможные значения: 0 – BASE-64 1 - DER
encryptAlgorithm	number	Необязательный параметр. Алгоритм шифрования. Возможные значения: 0 – ГОСТ 28147-89 1 – ГОСТ 34.12-2015 Магма 2- ГОСТ 34.12-2015 Кузнечик
token	string	Необязательный параметр. Токен, который будет использоваться при скачивании файлов с сервиса (параметр запроса)
tspURL	string	Необязательный параметр. Адрес службы штампов времени

ocspURL	string	Необязательный параметр. Адрес службы актуальных статусов

2.3.8. Интерфейс IDirectResults

Объекты данного типа описывают результаты прямой операции.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
directResults	IDirectResultOut[]	Массив результатов прямых операций

2.3.9. Интерфейс IDirectResultOut

Объекты данного типа содержат результаты прямой операции для файла.

Свойство	Тип	Описание
id?	string	Необязательный параметр. Идентификатор исходного файла. Если включено архивирование, то данный параметр не используется, т.к. выходной файл один для всех исходных
out	string	Результат операции в BASE-64
signers?	ISignerStatus[]	Необязательный параметр. Сведения о подписчиках

2.3.10. Интерфейс IReverseResults

Объекты данного типа описывают результаты обратной операции.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
reverseResults	IReverseResultOut[]	Массив результатов обратных операций

2.3.11. Интерфейс IReverseResultOut

Объекты данного типа описывают результаты обратной операции.

Свойство	Тип	Описание
id	string	Необязательный параметр. Идентификатор исходного файла
out	string	Результат операции в BASE-64

2.3.12. Интерфейс IVerifySignResults

Объекты данного типа описывают результаты проверки подписи.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
verifyResults	IVerifySignResult[]	Массив результатов проверки

2.3.13. Интерфейс IVerifySignResult

Объекты данного типа описывают результаты проверки подписи.

Свойство	Тип	Описание
id	string	Идентификатор исходного файла
status	boolean	Общий статус проверки подписи
signers	ISignerStatus[]	Информация о подписчиках документа

2.3.14. Интерфейс ISignerStatus

Объекты данного типа описывают сведения о подписчиках.

Свойство	Тип	Описание
signerCertificate	ICertificateInfo	Сведения о сертификате подписчика
status	boolean	Статус подписи

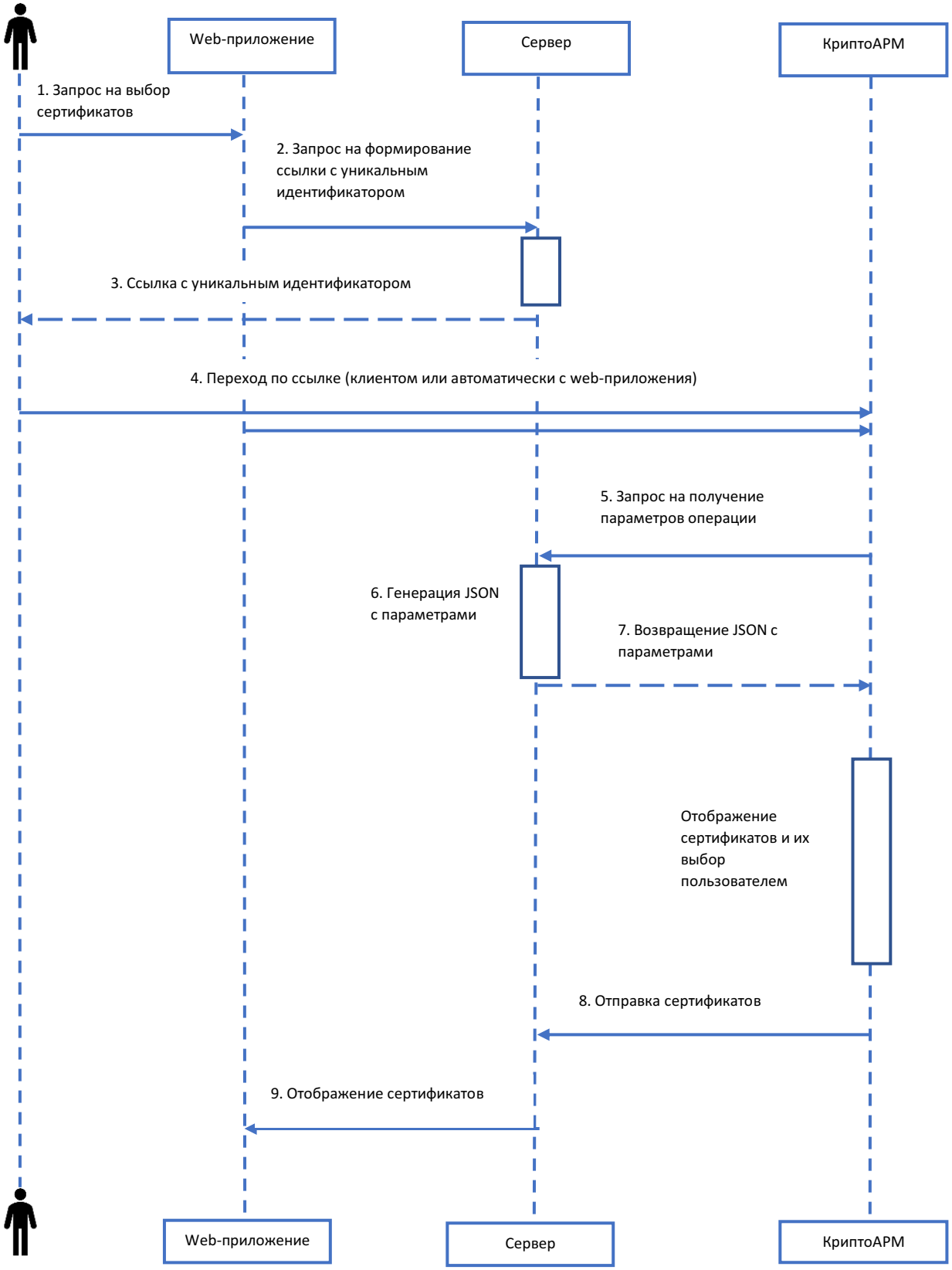
2.4. Интерфейс КриптоАРМ при подписи документов

При выполнении операции подписи по ссылке, не относящийся к процедуре интерфейс блокируется. Пользователю доступны: выбор сертификата, часть настроек подписи. Кнопка «Выполнить» заменяется двумя: «Подпись» и «Отмена». После выполнения команды, приложение будет свернуто в системный трей.

3. Команда certificates. Запросы на экспорт или импорт сертификатов и просмотр информации о сертификате

Команда **certificates** используется для: экспорта сертификата или списка сертификатов, а импорта сертификата в локальное хранилище, просмотра информации о сертификате.

Схема взаимодействия (экспорт сертификатов):



3.1. Формат ссылки

Для выполнения команды certificates должна быть сформирована ссылка вида:

cryptoarm://certificates/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **certificates** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://certificates/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

3.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе 1. Описание запросов и ответов.

3.2.1. Получение параметров операции

После получения команды **certificates** КриптоАРМ отправляет запрос на получение параметров операции.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.parameters»	Используемый метод. Всегда «certificate.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
diagnostic	IDiagnosticInformaton	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certificates.parameters",
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
     }
```

```
}
```

Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	ICertificatesParameters	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)

Пример ответа для экспорта сертификата:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": "export",
        "props": {
            "store": ["MY"],
            "multy": false
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

Пример ответа для импорта сертификата:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": "import",
        "props": {
```

```
            "store": ["MY"],
            "certificateBase64": "MIIFFDCCBMGgAwIBAgIQT...4VVkDWbX/n4="
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

Пример ответа для просмотра информации о сертификате:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": "information",
        "props": {
            "certificateBase64": "MIIFFDCCBMGgAwIBAgIQT...4VVkDWbX/n4="
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

3.2.2. Отправка сертификата

При экспорте сертификатов результат отправляются на сервер. После того, как пользователь выберет нужный сертификат КриптАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.base64»	Используемый метод. Всегда «certificates.base64»
params	ICertificateBase64Params	Параметры, содержащие объект сертификата

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
```

```
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certificates.base64",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "certificateBase64": "MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
        "friendlyName": "Минкомсвязь России"
    }
}
```

3.2.3. Отправка списка сертификатов

При экспорте сертификатов результат отправляются на сервер. Если параметры ICertRequestParameters содержат поле «multy» со значением «true», то пользователю в КриптоАРМ будет разрешён множественный выбор сертификатов. После того, как пользователь выберет нужные сертификаты и нажмет кнопку Готово, КриптАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не используется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.base64»	Используемый метод. Всегда «certificates.base64»
params	ICertificateBase64Params[]	Параметры запроса

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certificates.base64",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "certificates": [{
                "certificateBase64":
"MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
                "friendlyName": "Минкомсвязь России"
            },
            {
                "certificateBase64":
"MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
                "friendlyName": "Головной удостоверяющий центр"
```

```
            }
        ]
    }
}
```

3.2.4. Отправка сведений о сертификате

После того, как пользователь выберет нужный сертификат КриптАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.information»	Используемый метод или вид команды. Всегда «certificates.information»
params	Объект типа ICertificateInfo	Сведения о сертификате

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certificates.information",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "hash": "MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV",
        "issuerFriendlyName": "Минкомсвязь России",
        "issuerName": "Минкомсвязь России",
        "subjectFriendlyName ": "Минкомсвязь России",
        "subjectName": "Минкомсвязь России",
        "status": true
    }
}
```

3.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **certificates.**

3.3.1. Интерфейс ICertificatesParameters

Объекты данного типа описывают параметры команды запроса на получение сертификат

Свойство	Тип	Описание

operation	string	Тип операции импорт, экспорт, информация. Доступные значения: “import”, “export”, “information”
props	ICertificatesOperationProps	Параметры операции

3.3.2. Интерфейс ICertificatesOperationProps

Объекты данного типа описывают дополнительные свойства операции.

Свойство	Тип	Описание
headerText?	string	Необязательные параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательные параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов
store?	string[]	Необязательный параметр. Массив имен хранилищ. Если не задан, то используется значение MY (личные сертификаты)
multy?	boolean	Необязательный параметр. Разрешен ли множественный выбор. По умолчанию false
certificateBase64?	string	Необязательный параметр. Сертификат в формате X.509 закодированный в Base64

3.3.3. Интерфейс ICertificateBase64Params

Объекты данного типа описывают параметры запроса для метода certificates.base64.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
certificateBase64	string	Сертификат в формате X.509 закодированный в Base64
friendlyName	string	Дружественное имя сертификата

3.3.4. Интерфейс ICertificateInfo

Объекты данного типа описывают объекты, содержащие свойства сертификата

Свойство	Тип	Описание
id	string	Идентификатор транзакции
hash	string	SHA1 отпечаток
issuerFriendlyName	string	Дружественное имя издателя (CN)
issuerName	string	Имя издателя
notAfter	string	Дата окончания действия сертификата
notBefore	string	Дата начала действия сертификата
rootCA MinComSvyaz	boolean	Флаг, обозначающий является ли владельцем корневого сертификата цепочки “Минкомсвязь России”

subjectFriendlyName	string	Дружественное имя субъекта (CN)
subjectName	string	Имя субъекта
status	boolean	Статус сертификата. Проверяется вся цепочка
serial	string	Серийный номер сертификата
x509?	string	Необязательный параметр. Сертификат в формате X.509 закодированный в Base64

3.3.5. Интерфейс ICertificateIdentityInfo

Объекты данного типа описывают объекты, содержащие параметры сертификата, позволяющие только идентифицировать его.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
hash	string	SHA1 отпечаток
rootCA MinComSvyaz	boolean	Флаг, обозначающий является ли владельцем корневого сертификата цепочки “Минкомсвязь России”
status	boolean	Статус сертификата. Проверяется вся цепочка
pubKeyAlg	string	OID алгоритма открытого ключа сертификата

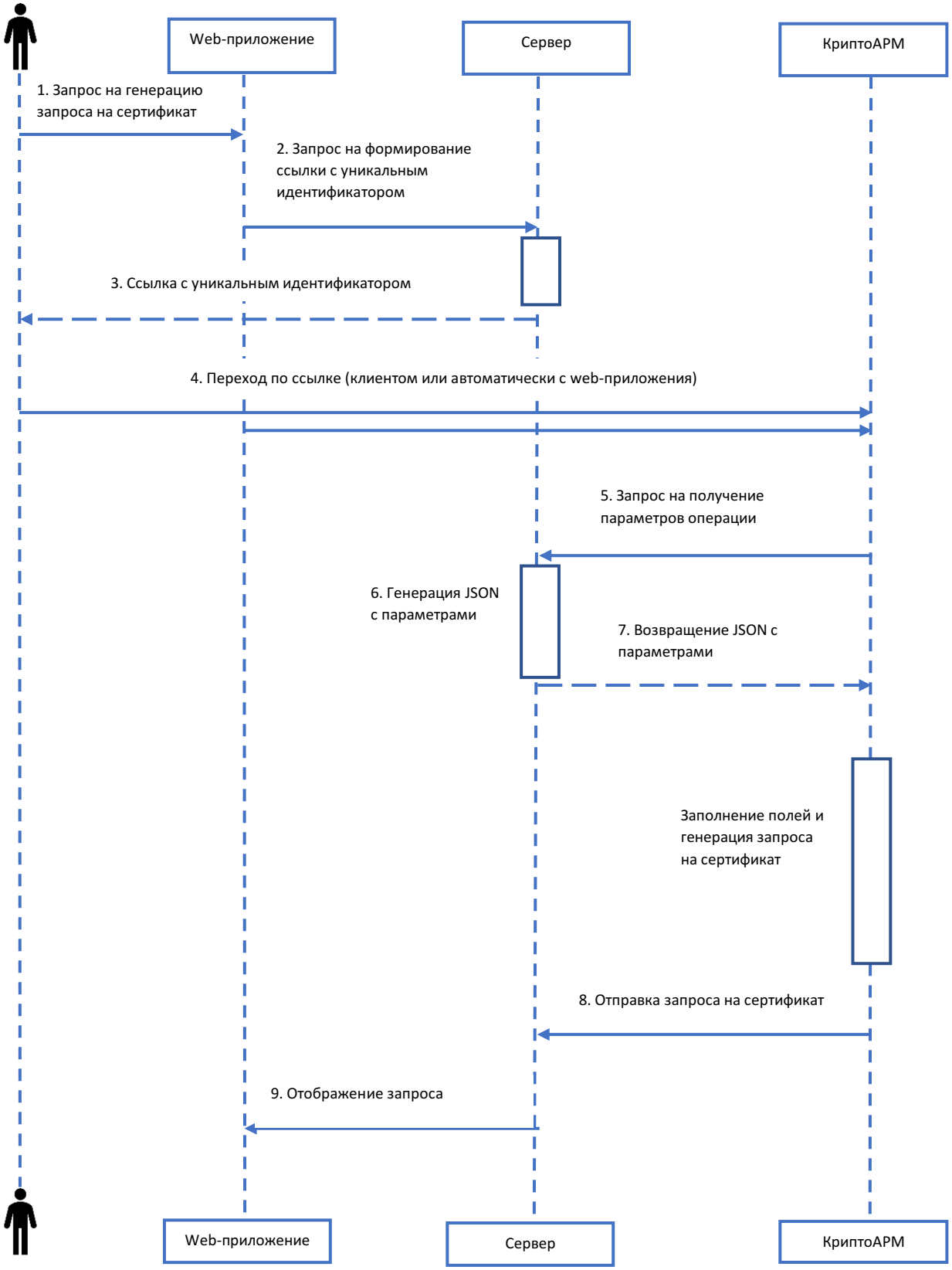
3.4. Интерфейс КриптоАРМ при выборе и отправке сертификатов

При выполнении команды запроса на сертификаты, не относящийся к процедуре интерфейс блокируется. Пользователю доступны: выбор сертификатов. Доступны две кнопки: «Готово» и «Отмена». После выполнения команды, приложение будет свернуто в системный трей.

4. Команда certrequests. Генерация запросов на сертификат

Команда **certrequests** используется для: генерации запроса на сертификат по шаблону или экспорт запросов

Схема взаимодействия (экспорт сертификатов):



4.1. Формат ссылки

Для выполнения команды certrequests должна быть сформирована ссылка вида:

cryptoarm://certrequests/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **certrequests** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://certrequests/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

4.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе 1. Описание запросов и ответов.

4.2.1. Получение параметров операции

После получения команды **certrequests** КриптоАРМ отправляет запрос на получение параметров операции.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certrequests.parameters»	Используемый метод. Всегда «certrequests.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
diagnostic	IDiagnosticInformaton	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certrequests.parameters",
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
    }
}
```

Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	ICertrequestsParameters	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)

Пример ответа:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": "GENERATE",
        "props": {
            "templateType": "JSONTemplate",
            "template": {
                "Description": "",
                "FriendlyName": "Пользователь",
                "RDN": [{
                    "Oid": "2.5.4.3",
                    "Name": "CN",
                    "Length": 64,
                    "LocalizedName": "Общее имя",
                    "SettingsValues": [],
                    "DefaultValue": null,
                    "ProhibitAnyValue": false,
                    "ProhibitChange": false,
                    "ProhibitEmpty": true
                }],
                "Extensions": {
                    "KeyUsage": [{
                        "Name": "cRLSign",
                        "LocalizedName": "Автономное подписание списка отзыва
(CRL)",
                        "DefaultValue": false,
                        "ProhibitChange": true
                    }],
```

```
                    "ExtendedKeyUsage": [{
                        "Name": "1.3.6.1.5.5.7.3.1",
                        "LocalizedName": "Проверка подлинности сервера",
                        "DefaultValue": false,
                        "ProhibitChange": true
                    }]
                },
                "MarkExportable": false
            },
            "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
        }
    }
}
```

4.2.2. Отправка запроса на сертификат

После генерации запроса, он будет отправлен на сервер. Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certrequests.base64»	Используемый метод. Всегда «certrequests.base64»
params	ICertificaterequestBase64Params	Параметры, содержащие объект сертификата

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "certrequests.base64",
    "params": {
        "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
        "certificaterequestBase64":
"MIIFFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
        "friendlyName": "ООО Рога и Копыта"
    }
}
```

4.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **certrequests.**

4.3.1. Интерфейс ICertrequestsParameters

Объекты данного типа описывают параметры команды.

Свойство	Тип	Описание
operation	string	Тип операции: генерация запроса на сертификат. Значение типа: CertrequestsOperation
props	ICertrequestsOperationGenerateProps	Параметры операции

4.3.2. Тип CertrequestsOperation

Возможные операции с запросами на сертификат.

Значение	Описание
GENERATE	Генерация запроса на сертификат

4.3.3. Интерфейс ICertrequestsOperationGenerateProps

Объекты данного типа описывают дополнительные свойства операции.

Свойство	Тип	Значение	Описание
headerText?	string	Необязательные параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов	headerText?
descriptionText?	string	Необязательные параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов	descriptionText?
templateType	string	JSONTemplate	В качестве шаблона используется JSON типа IJSONTemplate
		CertificateTemplate	В качестве шаблона используется сертификат
template	IJSONTemplate или ICertificateTemplate	Шаблон для генерации запроса на сертификат	

4.3.4. Интерфейс IJSONTemplate

Объекты данного типа описывают поля для генерации запроса на сертификат.

Свойство	Тип	Описание
Description	string	Описание шаблона
FriendlyName	string	Дружественное имя шаблона
RDN	IRDN[]	Набор полей DN
Extensions	IRequestExtension	Расширения

MarkExportable	boolean	Определяет экспортируемость ключей

4.3.5. Интерфейс IRDN

Свойство	Тип	Описание
Oid	string	Описание шаблона
Name	string	Наименование OID
Length	number	Максимальная длина поля
LocalizedName	string	Локализованное наименование OID
SettingsValues	string[]	Список возможных значений
DefaultValue	string	Значение по умолчанию
ProhibitAnyValue	boolean	Флаг указывающий, что пользователю доступны только значения из массива SettingsValues
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено
ProhibitEmpty	boolean	Флаг указывающий, что поле должно быть непустым

4.3.6. Интерфейс IRequestExtension

Свойство	Тип	Описание
KeyUsage	IKeyUsage[]	Описание шаблона
ExtendedKeyUsage	IExtendedKeyUsage[]	Наименование OID

4.3.7. Интерфейс IKeyUsage

Свойство	Тип	Описание
Name	string	Наименование OID
LocalizedName	string	Локализованное наименование OID
DefaultValue	boolean	Значение по умолчанию
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено

4.3.8. Интерфейс IExtendedKeyUsage

Свойство	Тип	Описание
Name	string	Наименование OID
LocalizedName	string	Локализованное наименование OID
DefaultValue	boolean	Значение по умолчанию
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено

4.3.9. Интерфейс ICertificateTemplate

Объекты данного типа используются для генерации запроса на сертификат, где запрос формируется по сертификату.

Свойство	Тип	Описание
certificateBase64	string	Сертификат в формате X.509 закодированный в Base64

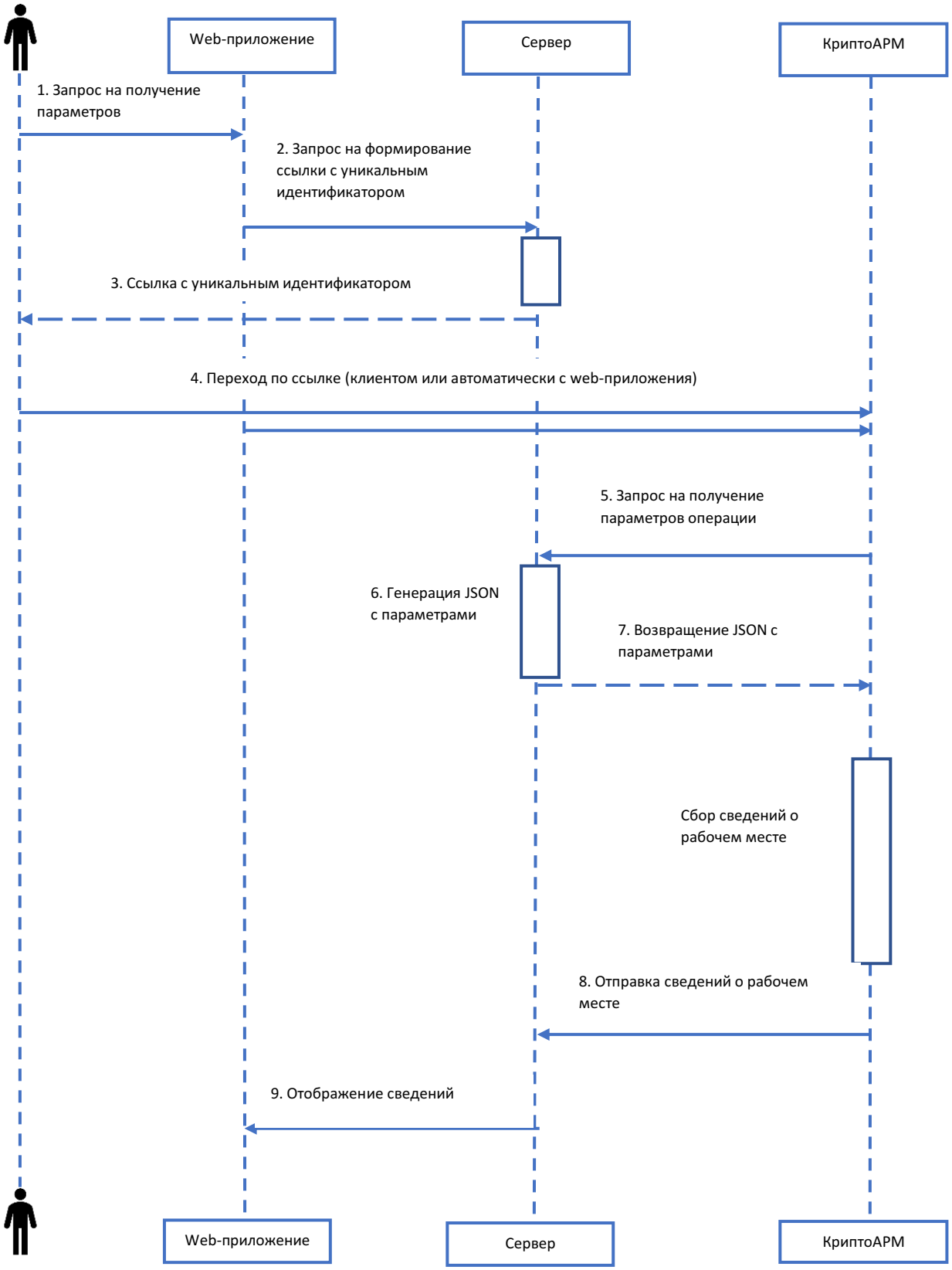
4.3.10. Интерфейс ICertificaterequestBase64Params

Объекты данного типа описывают параметры запроса для метода certrequests.base64.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
certificaterequestBase64	string	Запрос в формате X.509 закодированный в Base64
friendlyName	string	Дружественное имя субъекта

5. Команда diagnostics. Запросы на диагностику рабочего места

Команда **diagnostics** используется для диагностики рабочего места пользователя. Схема взаимодействия:



5.1. Формат ссылки

Для выполнения команды diagnostics должна быть сформирована ссылка вида:

cryptoarm://diagnostics/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **diagnostics** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://diagnostics/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

5.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе 1. Описание запросов и ответов.

5.2.1. Получение параметров операции

После получения команды **diagnostics** КриптоАРМ отправляет запрос на получение параметров операции.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«diagnostics.parameters»	Используемый метод. Всегда «diagnostics.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
diagnostic	IDiagnosticInformaton	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "diagnostics.parameters",
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
    }
```

```
}
```

Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	IDiagnosticsParameters	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)

Пример ответа:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        "operation": ["CSP_ENABLED", "LICENSES", "VERSIONS"],
        "props": {
            "headerText": "Диагностика cryptoarm.ru",
            "descriptionText": "Выполняется диагностика рабочего места для работы
на портале"
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

5.2.2. Отправка сведений о рабочем месте

Полученные сведения отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«diagnostics.information»	Используемый метод. Всегда «diagnostics.information»
params	Объект типа IDiagnosticsInformation	Сведения о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
  "jsonrpc": "2.0",
  "method": "diagnostics.information",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "CSP_ENABLED": true,
    "LICENSES": {
      "csp": {
        "status": true,
      },
      "cryptoarm": {
        "status": true,
        "type": "temporary",
        "expiration": "1591689487056",
      }
    },
    "VERSIONS": {
      "csp": "5.0.11753",
      "cryptoarm": "2.5.2",
    }
  }
}
```

5.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **diagnostics.**

5.3.1. Интерфейс IDiagnosticsParameters

Объекты данного типа описывают вид операции и её параметры

operation	string[]	Тип операции. Доступные значения типа: IDiagnosticOperation
props	IDiagnosticsOperationProps	Параметры операции

5.3.2. Тип IDiagnosticOperation

Возможные операции.

Значение	**Описание**
SYSTEMINFORMATION	Сведения о системе
CSP_ENABLED	Наличие КриптоПро CSP
CADES_ENABLED	Доступность CADES
VERSIONS	Версии используемых компонентов (КриптоАРМ, КриптоПро)

PROVIDERS	Список криптопровайдеров
LICENSES	Статус лицензий
PERSONALCERTIFICATES	Наличие личных сертификатов

5.3.3. Интерфейс IDiagnosticsOperationProps

Интерфейс IDiagnosticsOperationProps описывает параметры операции.

Свойство	Тип	Описание
headerText?	string	Необязательные параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательные параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов

5.3.4. Интерфейс IDiagnosticsInformation

Объекты данного типа описывают объекты, содержащие сведения о рабочем месте

Свойство	Тип	Описание
id	string	Идентификатор транзакции
API_VERSION	string	Версия поддерживаемого КриптоАРМ API
SYSTEMINFORMATION	ISystemInformation	Сведения о системе
CSP_ENABLED	boolean	Установлен или нет КриптоПро CSP
CADES_ENABLED	boolean	Доступность CADES
VERSIONS	IVersions	Версии компонентов
PROVIDERS	IProviders	Сведения о провайдерах
LICENSES	ILicenses	Сведения о лицензиях
PERSONALCERTIFICATES	ICertificateIdentityInfo[]	Сведения о личных сертификатах (только идентификаторы)

5.3.5. Интерфейс ISystemInformation

Объекты данного типа содержат сведения о системе пользователя.

Свойство	Тип	Описание
type	string	Тип системы. Возможные значения: 'Linux', 'Darwin' и 'Windows_NT'
arch	string	Архитектура операционной системы. Возможные значения: 'arm', 'arm64', 'ia32', 'mips', 'mipsel', 'ppc', 'ppc64', 's390', 's390x', 'x32', и 'x64'
platform	string	Имя платформы. Возможные значения: 'aix', 'darwin', 'freebsd', 'linux', 'openbsd', 'sunos', и 'win32'
packageType?	string	Необязательный параметр. Тип поддерживаемого пакета (инсталлятора).

		Возможные значения: ‘msi’, ‘pkg’, ‘rpm’ или ‘deb’

5.3.6. Интерфейс IVersions

Объекты данного типа содержат сведения о версиях.

Свойство	Тип	Описание
csp	string	Версия КриптоПро CSP
cryptoarm	string	Версия КриптоАРМ

5.3.7. Интерфейс IProviders

Объекты данного типа описывают доступность ГОСТ провайдеров.

Свойство	Тип	Описание
GOST2012_256	boolean	ГОСТ 2012-256
GOST2012_512	boolean	ГОСТ 2012-512

5.3.8. Интерфейс ILicenses

Объекты данного типа описывают статусы лицензии КриптоАРМ и КритпоПро CSP.

Свойство	Тип	Описание
csp	ILicenseInfo	Сведения о лицензии на КриптоПро CSP
cryptoarm	ILicenseInfo	Сведения о лицензии на КриптоАРМ

5.3.9. Интерфейс ILicenseInfo

Объекты данного типа описывают сведения о лицензии компонента.

Свойство	Тип	Описание
status	boolean	Действительна или нет лицензия
type	LicenseType	Тип лицензии.
expiration?	string	Необязательный параметр. Дата истечения лицензии для трилальных лицензий или подписок (в формате UTC)

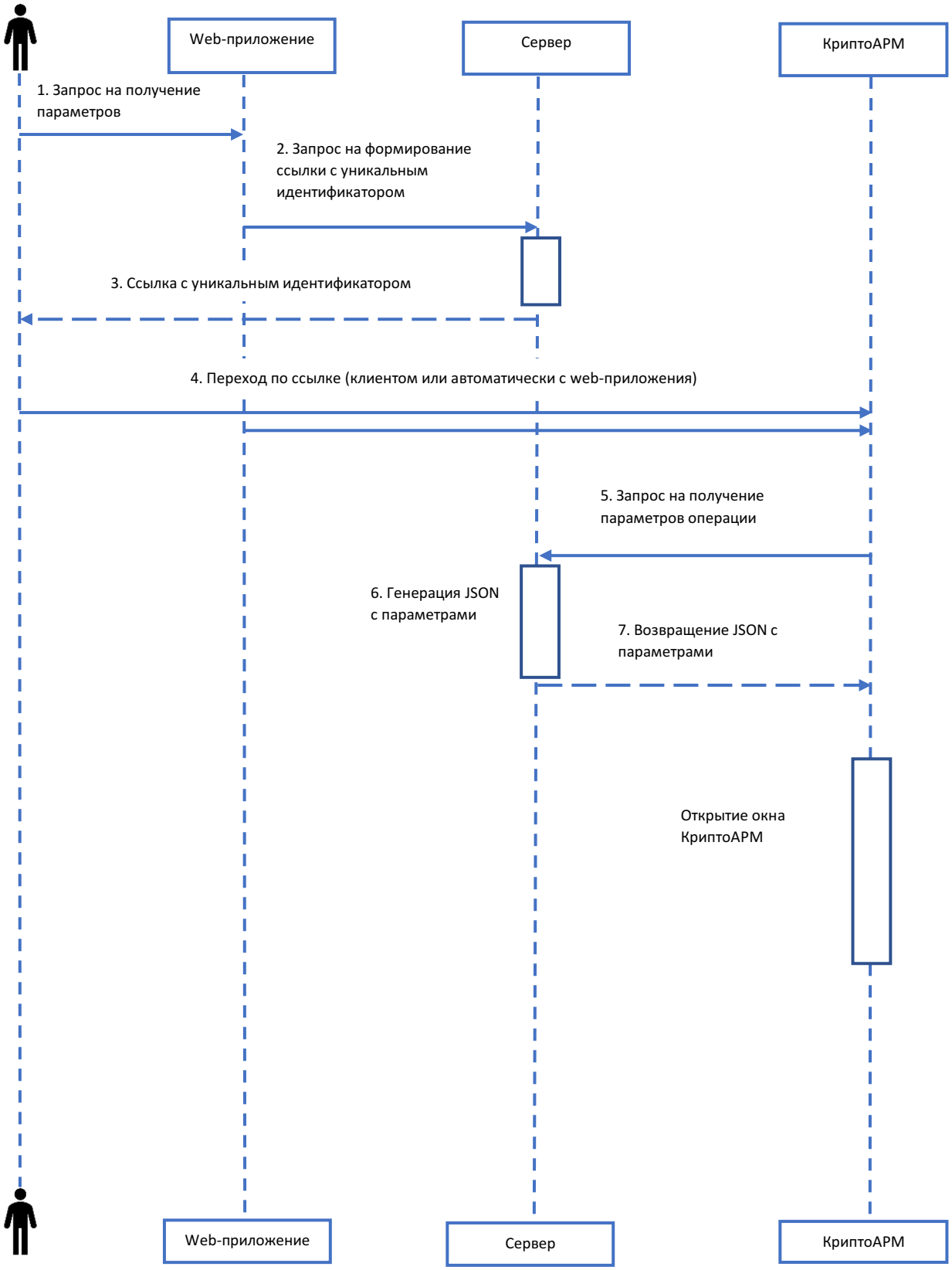
5.3.10. LicenseType Enum

Данное перечисление описывает возможные типы лицензий

Значение	Описание
Permanent	Постоянная
Subscription	Подписка
Daily	Дневная
Trial	Триальная

6. Команда startView. Открытие окна приложения

Команда **startView** используется для открытия конкретного окна приложения КриптоАРМ ГОСТ. Схема взаимодействия:



6.1. Формат ссылки

Для выполнения команды diagnostics должна быть сформирована ссылка вида:

cryptoarm://startView/<URL>/?id=<id>

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **startView**- выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://startView/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

6.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе 1. Описание запросов и ответов.

6.2.1. Получение параметров операции

После получения команды **startView** КриптоАРМ отправляет запрос на получение параметров операции.

Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«startView.parameters»	Используемый метод. Всегда «startView.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)
diagnostic	IDiagnosticInformaton	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
    "jsonrpc": "2.0",
    "method": "startView.parameters",
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
    }
```

```
}
```

Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	IStartViewParameters	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию («Формат ссылки»)

Пример ответа:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
    "jsonrpc": "2.0",
    "result": {
        " uiView": “CERTIFICATES_MY”,
        "props": {
            "headerText": "ИС cryptoarm.ru",
            "descriptionText": "Запрос на открытие окна"
        }
    },
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
```

6.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **startView.**

6.3.1. Интерфейс IStartViewParameters

Объекты данного типа описывают вид операции и её параметры

uiView	string	Тип окна, которое нужно отобразить пользователю. Доступные значения: “MAIL” – окно почты “DOCUMENTS” – окно Документы “SIGN_AND_ENCRYPT” - окно подписи и шифрования

		“CERTIFICATES_MY” - хранилище личных сертификатов “CERTIFICATES_ADDRESS_BOOK” “CERTIFICATES_CA” “CERTIFICATES_ROOT” “KEYS” - окно ключей “ABOUT” - окно "О программе" “DIAGNOSTIC_PROBLEM_PERSONAL_CERTIFICATES” – окно диагностики, с проблемой отсутствия личных сертификатов
props	IStartViewOperationProps	Параметры операции

6.3.2. Интерфейс IStartViewOperationProps

Интерфейс IStartViewOperationProps описывает параметры операции.

Свойство	Тип	Описание
headerText?	string	Необязательные параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательные параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов