

Thrift module: client_api

Quantum key distribution Thrift API.

Русская версия представлена ниже.

Quantum key API of device consists of two functions: get new quantum key and it's unique QID from device by length and get existing (already requested on paired device) key by it's QID.

Quantum key use routine.

QKD pair consists of two paired devices. We'll call one side as "A" and other one as "B".

1. Side "A"

- a. Key consumer on side "A" requests new quantum key from device (on side A) providing length of new key.
- b. Device on side "A" responds with new quantum key and corresponding QID for this key.

2. Transmission from side "A" to side "B"

- a. Key consumer on side "A" sends QID to key consumer on side "B".

3. Side "B"

- a. Key consumer on side "B" requests existing quantum key from device (on side B) providing QID received from side "A".
- b. Device on side "B" responds with same quantum key as device on side "A" (for given QID).

Device behaviour on side "A" and side "B" is absolutely same for key consumers.

API server needs SSL (TLSv1.2) socket to be used for client connections. Client must provide X509 certificate for authentication.

Binary serialization protocol is used for Thrift connections.

API versioning implemented by thrift's namespaces. Backward-compatible API versions use the same namespaces.

Thrift-api для получения квантовых ключей.

API включает в себя две функции для получения квантовых ключей от API-сервера (квантового устройства): получение по длине и получение по идентификатору (квиду).

Последовательность действий по получению и использованию одного ключа.

Назовем одну из сторон канала передачи данных "А", другую – "Б".

1. Сторона «А»

- a. Потребитель на стороне «А» запрашивает квантовый ключ требуемой ему длины у квантового устройства на своей стороне.
- b. Квантовое устройство на стороне «А» возвращает потребителю квантовый ключ требуемой длины вместе с идентификатором (квидом) этого ключа.

2. Передача от «А» к «Б»

- a. Потребитель на стороне «А» передает квид потребителю на стороне «Б».

3. Сторона «Б»

- a. Потребитель стороне «Б», получив квид ключа, запрашивает у квантового устройства на своей стороне квантовый ключ, соответствующий полученному от «А» квид-у.
- b. Квантовое устройство на стороне "Б" возвращает тот же ключ, что был возвращен (с указанным квид-ом) на стороне "А".

С точки зрения потребителя сторона "А" и сторона "Б" ничем не отличаются и предоставляют одинаковый интерфейс.

Для подключения к серверу API необходимо использовать SSL (TLSv1.2) сокет с обязательной передачей клиентского X509 сертификата.

Используется Binary протокол сериализации.

Версионирование при сохранении обратной совместимости не требуется (добавление полей и т.д.). Несовместимые версии API используют другой namespace (или его аналог в целевом языке).

Module	Services	Data types	Constants
client_api	ClientApiService <ul style="list-style-type: none">get_by_idget_by_length	CLIENT_ERROR_CODE ClientError KeyInfo SERVER_ERROR_CODE ServerError	

Enumerations

Enumeration: SERVER_ERROR_CODE

Server side error codes (ServerError). Коды серверных ошибок (ServerError).

ERROR_BUSY	-1	Server busy or overloaded. Retry later. Сервер занят или перегружен, необходимо повторить запрос позднее.
ERROR_KEY_EXHAUSTED	-2	Insufficient key data on server for processing key request. Retry later. Недостаточно накопленного ключа на сервере, необходимо повторить запрос позднее.
ERROR_INTERNAL	-99	Internal server error. Внутренняя ошибка сервера.

Enumeration: CLIENT_ERROR_CODE

Client side error codes (ClientError). Коды клиентских ошибок (ClientError).

ERROR_KEY_UNKNOWN	-101	Wrong QID provided (non-existent, already used or expired). Указан неверный идентификатор ключа (несуществующий, использованный или истёкший).
ERROR_INVALID_ARGUMENT	-102	Invalid parameter value (usually key length). Указано недопустимое значение параметра.

Data structures

Struct: KeyInfo

Key	Field	Type	Description	Requiredness	Default value
1	key_body	binary	Key body, always present. Тело ключа, возвращается всегда.	default	

Key	Field	Type	Description	Requiredness	Default value
2	key_id	binary	Key identifier, filled on get_by_length() calls, 16 bytes Must be used for calling get_by_id() on paired device. Идентификатор ключа, возвращается при вызове get_by_length(), 16 байт Используется для последующего получения этого ключа на принимающей стороне.	default	
3	expiration_time	i64	Key lifetime, filled on get_by_length() calls. UNIX timestamp in millis, UTC+0 timezone. Время действия ключа, возвращается при вызове get_by_length(). Представлено как UNIX timestamp в миллисекундах в зоне UTC+0.	default	

Key data structure. Returned for all key requests. Структура информации о ключе, возвращается при всех запросах на получение ключа.

Exception: ServerError

Key	Field	Type	Description	Requiredness	Default value
1	error_code	i32	Error code. Код ошибки.	default	
2	retry_after	double	Time amount before retrying request. Время, через которое можно повторить запрос.	default	
3	message	string	Error description. Текстовое описание ошибки.	default	

Server side error, can't be fixed by client. Client must wait for retry_after seconds and retry the same request. Ошибка на стороне сервера, клиент не может на неё повлиять. Можно подождать retry_after секунд и повторить запрос.

Exception: ClientError

Key	Field	Type	Description	Requiredness	Default value
1	error_code	i32	Error code. Код ошибки.	default	
2	message	string	Error description. Текстовое описание ошибки.	default	

Client side error (wrong request). Retrying with same request will cause same error. Ошибка на стороне клиента. Повторный запрос приведет к этой же ошибке.

Services

Service: ClientApiService

Function: ClientApiService.get_by_length

```
KeyInfo get_by_length(i32 key_length)
    throws ServerError, ClientError
```

Get new key and it's QID for given key length. Получить новый ключ указанной длины и его идентификатор.

Parameters

Name	Description
key_length	Key length in bytes. Длина запрашиваемого ключа в байтах.

Function: ClientApiService.get_by_id

```
KeyInfo get_by_id(binary key_id)  
    throws ServerError, ClientError
```

Get existing key by it's QID. Получить существующий ключ по его идентификатору.

Parameters

Name	Description
key_id	Key identifier (QID, 16 bytes). Value determined in KeyInfo structure, returned for get_by_length() request. Идентификатор ключа (квид, 16 байт), указанный в структуре KeyInfo при получении.