Клиент генерирует ephemeral ECDHE ключ и отправляет серверу.

|  |  |  |
| --- | --- | --- |
| ClientHello | Генерация ECDHE ключа | \_client\_ecdh.get\_public\_key\_der() |
|  | Отправка ключа | UtilsNetwork::write\_uint32, boost::asio::write() в TLSClient::run\_handshake\_and\_send() |

Сервер генерирует свой ECDHE ключ, отправляет клиенту и производит обмен ключами.

|  |  |  |
| --- | --- | --- |
| ServerHello | Генерация ECDHE ключа сервера | \_server\_ecdh.get\_public\_key\_der() |
|  | Отправка ключа | UtilsNetwork::write\_uint32, boost::asio::write() в TLSServer::handle\_handshake() |

Обе стороны вычисляют общий секрет.

|  |  |  |
| --- | --- | --- |
| KeyShare (client) | Получение server\_pub | boost::asio::read() в TLSClient |
|  | Вычисление shared\_secret | \_client\_ecdh.compute\_shared\_secret() |
| KeyShare (server) | Получение client\_pub | boost::asio::read() в TLSServer |
|  | Вычисление shared\_secret | \_server\_ecdh.compute\_shared\_secret() |

Оба участника вычисляют симметричный ключ из общего секрета.

|  |  |  |
| --- | --- | --- |
| derive secrets | SHA-256 от shared\_secret | UtilsCrypto::sha256(shared\_secret) |
|  |  |  |
| установка AES ключа | Установка ключа | AESCrypto aes(aes\_key) |
|  |  |  |

Клиент шифрует сообщение и отправляет серверу. Сервер расшифровывает.

|  |  |  |
| --- | --- | --- |
| ApplicationData (client) | Шифрование AES | aes.encrypt() |
|  | Отправка IV, tag, ciphertext | boost::asio::write() |
| ApplicationData (server) | |  | | --- | |  |  |  | | --- | | Чтение IV, tag, ciphertext | | boost::asio::read() |
|  | Расшифровка | aes.decrypt() |