SSL with ssl context

> This program allows only TLS 1.2 with ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256 only
>
> All other sslv1,v1, tlsv1.1 and 1.3 are disabled

```
#include "crow.h"
#include <sstream>
#include <openssl/ssl.h>

int main()
{
    crow::SimpleApp app; //define your crow application

    // Route handler for root path
    CROW_ROUTE(app, "/")([](){
        return "Hello world! (HTTPS Enabled)";
    });

    // Route handler with parameter
    CROW_ROUTE(app, "/hello/<int>")
    ([](int count){
        if (count > 100)
            return crow::response(400);
        std::ostringstream os;
        os << count << " bottles of beer!";
        return crow::response(os.str());
    });

    // Route handler for about page
    CROW_ROUTE(app, "/about")([](){
        return "This is a Crow C++ web framework example with HTTPS
support!";
    });

    // Create and configure SSL context to use only TLS 1.2
    asio::ssl::context ssl_ctx(asio::ssl::context::tlsv12);

    // Set certificate files
    ssl_ctx.use_certificate_file("server.crt", asio::ssl::context::pem);
    ssl_ctx.use_private_key_file("server.key", asio::ssl::context::pem);
```

```
    // Configure TLS 1.2 only
    SSL_CTX_set_min_proto_version(ssl_ctx.native_handle(), TLS1_2_VERSION);
    SSL_CTX_set_max_proto_version(ssl_ctx.native_handle(), TLS1_2_VERSION);

    // Set specific cipher suites for TLS 1.2
    SSL_CTX_set_cipher_list(ssl_ctx.native_handle(), "ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-RSA-AES128-GCM-SHA256");

    // Set SSL options
    ssl_ctx.set_options(
        asio::ssl::context::default_workarounds |
        asio::ssl::context::no_sslv2 |
        asio::ssl::context::no_sslv3 |
        asio::ssl::context::no_tlsv1 |
        asio::ssl::context::no_tlsv1_1 |
        asio::ssl::context::no_tlsv1_3
    );

    // Start the server with HTTPS enabled on port 8443
    app.port(8443)
       .ssl(std::move(ssl_ctx))  // Use our custom SSL context
       .multithreaded()
       .bindaddr("127.0.0.1").port(8443)
       .run();

    return 0;
}
```

---

test using

> test that it works with TLS 1.2:

```
# echo "Q" | openssl s_client -connect 127.0.0.1:8443 -tls1_2 2>&1 | grep -E
"(Protocol|Cipher|TLS|New,)"
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
```

> test that TLS 1.3 is rejected:

```
# echo "Q" | openssl s_client -connect 127.0.0.1:8443 -tls1_3 2>&1 | grep -E "(alert|error|refused|Protocol|Cipher|TLS|New,|CONNECTED)"
(2025-07-27 17:55:16) [ERROR   ] Could not start adaptor: unsupported protocol (SSL routines)
40B7F5C12A730000:error:0A00042E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol version:../ssl/record/rec_layer_s3.c:1599:SSL alert number 70
CONNECTED(00000003)
New, (NONE), Cipher is (NONE)
```

> test that TLS 1.1 is rejected:

```
# echo "Q" | openssl s_client -connect 127.0.0.1:8443 -tls1_1 2>&1 | grep -E "(alert|error|refused|Protocol|Cipher|TLS|New,|CONNECTED)"
(2025-07-27 17:55:42) [ERROR   ] Could not start adaptor: unexpected message (SSL routines)4087E7D0C67C0000:error:0A0000BF:SSL routines:tls_setup_handshake:no protocols available:../ssl/statem/statem_lib.c:104:

CONNECTED(00000003)
New, (NONE), Cipher is (NONE)
```

> check the specific cipher suite being used:

```
# echo "GET / HTTP/1.0\r\n\r\n" | openssl s_client -connect 127.0.0.1:8443 -tls1_2 -cipher 'ECDHE-RSA-AES256-GCM-SHA384' 2>&1 | grep -E "(Cipher|Protocol|Server certificate)"
Server certificate
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
```

> test the alternative cipher suite we configured:

```
# echo "GET / HTTP/1.0\r\n\r\n" | openssl s_client -connect 127.0.0.1:8443 -tls1_2 -cipher 'ECDHE-RSA-AES128-GCM-SHA256' 2>&1 | grep -E "(Cipher|Protocol)"
New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
```

TLS Configuration:

- TLS Version: TLS 1.2 only (TLS 1.1, TLS 1.3 are rejected)

- Primary Cipher Suite: ECDHE-RSA-AES256-GCM-SHA384
- Alternative Cipher Suite: ECDHE-RSA-AES128-GCM-SHA256

Test Results:

1. TLS 1.2 connection: Works successfully with the configured cipher suites
2. TLS 1.3 connection: Rejected with “tlsv1 alert protocol version” error
3. TLS 1.1 connection: Rejected with “no protocols available” error
4. HTTPS requests: Server responds correctly to HTTPS requests

Key Configuration Changes:

- Created a custom asio::ssl::context with tlsv12 method
- Set minimum and maximum protocol versions to TLS 1.2 using OpenSSL Functions
- Configured specific cipher suites for TLS 1.2
- Disabled all other TLS/SSL versions (SSLv2, SSLv3, TLS 1.0, TLS 1.1, TLS 1.3)

> CMakeLists.txt

```
cmake_minimum_required(VERSION 3.10)

project(CrowApp)

# Set C++ standard
set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

# Find required packages
find_package(Crow CONFIG REQUIRED)
find_package(OpenSSL REQUIRED)
find_package(ZLIB REQUIRED)
find_package(Threads REQUIRED)

add_executable(CrowApp main.cpp)

target_link_libraries(CrowApp
    Crow::Crow
    OpenSSL::SSL
    OpenSSL::Crypto
    ZLIB::ZLIB
    Threads::Threads
)
```

Create ssl key and also copy them to build directory also