COLLABORATIVE WORK WITH AUGMENTED AND VIRTUAL REALITY - A SECURE NETWORK CONNECTION IN UNITY

By Lars Tijmsans

Doris Aschenbrenner (TU Delft)
Developing a system to host collaborative remote “visits” using augmented reality and virtual reality

Companies may send sensitive data over this connection

A secure connection is required!
2D/3D cross platform game engine
C++, C#
Focused on easy development

Latency > Security

UNET is deprecated. Successor not released yet

Mirror: Open source. Third party multiplayer library
THE PLAN

- Authenticate the user using AR/VR (Pin or pattern)
- Use TLS to establish a secure websocket connection between a server and a client in Unity
- Create a secure and “playable” client server setup
What is the performance of a TLS websocket connection in Unity compared to a standard insecure connection?
Step 1: Implement a test scenario that uses a websocket over TLS

Step 2: Create my own certificate authority (CA) and make it trusted by the server and the remote client

Step 3: Take measurements of the latency in the test scenario with TLS

Step 4: Take measurements of the latency in the test scenario without TLS

Step 5: Compile results and draw conclusions
Is this method suitable for secure gaming connections? -> Yes!

Secure: \(\frac{1000\text{ms}}{11.09\text{ms}} = 90\) updates per second

Insecure: \(\frac{100\text{ms}}{5.24\text{ms}} = 190\) updates per second

Is this the only implementation possible? -> No
FUTURE WORK

- Test the scenario on AR/VR glasses
  - They have less processing power than my testing machines
  - Modern CPUs are optimized for AES

- Implement different cipher suites
  - I used: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

- Unet successor may be a good alternative in the future

- Implement an authentication system
  - A user should prove it is him before being able to establish a connection
  - There needs to be a way to login (e.g. PIN or Pattern)