Exploring Encryption in Networked Multiplayer Games Developed in Unity3D

Pat Healy[1]

***Abstract*—**

I. INTRODUCTION

Cryptography is not a topic that typically intersects much with the world of video game development, even in networked multiplayer games. Much of the internet traffic that drives networked games is either minimally encrypted or just completely un-encrypted. There are a few reasons for that.

- Game data isn't sensitive.
- Games often require low latency.
- Implementing encryption can take time, which developers often do not want to spend.

A. Sensitive Data

When determining cryptographic schema, we should consider both that the cost of breaking the ciphertext exceeds the value of the encrypted information and the time required to break the ciphertext exceeds the useful lifetime of the information. Of course, sometimes games require the player input sensitive information, such as when logging in or registering for a gaming service; these cases should very obviously be encrypted in a secure manner but this is not the topic I'm discussing. Think instead of the basic moment-to-moment gameplay of an online multiplayer game.

In these scenarios, the data transmitted simply contains information about actions taken by the given player. For example, it could contain the locations of any game objects modified by the player and some kind of identification number to identify the player and tie the action to them. Or, in the other direction, the server sends the client information about other game objects that move outside of the player's direct actions.

*A project for INFSCI 2170 with Dr. Prashant Krishnamurthy.

[1]PhD Student in Information Science, University of Pittsburgh

This is not data of particularly high value, at least in most cases. An attacker can only use it (specifically, they may use any player id data that authenticates the user) to impact the outcome of the particular game the user is playing, either by impersonating the player in packets sent to the server or impersonating the server in packets sent back to the player. This may negatively impact the player's performance in a multiplayer game, but this is ultimately very low stakes, outside of professional competitive play. Still, a large scale set of attacks could negatively impact players' willingness to purchase and/or play the game.

The useful lifetime of the data is approximately the length of a given multiplayer game session. A single session of most online multiplayer games lasts less than an hour, easily. Just to be safe, I'd say the time to break the ciphertext should exceed five hours.

B. Latency

Here we see a divergence between turn-based and real-time multiplayer games. Latency is effectively not a problem with turn-based games; therefore, the run-time of encryption and decryption is not a concern. In real-time games, latency is a concern; players expect game objects to be updated at a rate of at least 30 frames per second. This is a complex issue, complicated by interpolation done on the part of the game developer outside of the exchange of packets (meaning transmission of 30 packets per second is not necessarily required), so instead of a quantitative measure of speed, it may make more sense to focus on whether players themselves can detect further latency.

C. Implementation

The labor required to implement encryption in multiplayer games is clearly not trivial. Given the

somewhat standard practice of not encrypting traffic at all, especially in the independently published video games made in Unity3D, implementation of a feasible cryptography schema will have to be incredibly easy and fast. Of course, this means taking advantage of existing encryption packages in the .NET framework.

D. The Plan

The goal of finding a feasible cryptographic schema for a multiplayer online game made in Unity3D is not to find the *best* algorithm but instead the easiest to implement algorithm that meets these requirements: (1) the addition of encryption should not introduce any human-noticeable latency, and (2) time to break ciphertext by brute force should exceed 5 hours.

This project is both an implementation of encrypted network traffic in a Unity3D multiplayer game and an assessment of several encryption algorithms within that developed framework.

REFERENCES

[1]