유니티 멀티플레이어 환경에서의 경량 암호 통신 구현

지도교수 : 손준영

팀 명 : 민트초코펩시

팀 원 : 201924409 고세화

201924438 김재민

201924521 이강인

목차

1. 연구 배경 및 목적

1.1 연구 배경

컴퓨터 산업이 발전함에 따라 컴퓨터 게임 산업도 급속도로 성장해 왔다. 초기에 단순한 콘솔 게임에서 시작된 게임 산업은 하드웨어 성능의 향상과 네트워크 기술의 발전에 힘입어 다양한 장르의 고품질 게임을 제공하는 산업으로 성장했다. 특히 인터넷의 보급과 함께 멀티플레이어 게임의 인기가 상승했으며, 네트워크 상에서 다른 사용자들과 실시간으로 게임을 즐길 수 있게 되었다. 이러한 환경 변화는 게임 개발에 있어 네트워크 통신 기술의 중요성을 부각시켰고, 유니티(Unity)와 같은 게임 엔진이 멀티플레이어 게임 개발을 위한 다양한 네트워킹 기능을 제공하는 계기가 되었다.

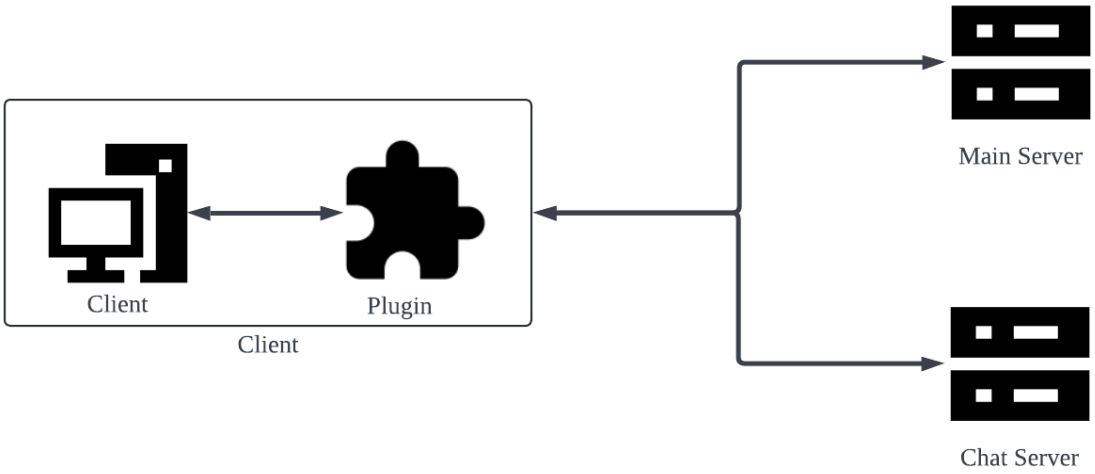
유니티는 전 세계적으로 널리 사용되는 게임 엔진으로, 다양한 플랫폼에서 멀티플레이어 게임을 개발할 수 있는 환경을 제공한다. 유니티는 멀티플레이어 게임 개발을 위해 다양한 네트워킹 기술을 지원한다. 하지만, 멀티플레이어 게임의 특성상 클라이언트와 서버 간의 통신이 빈번하게 이루어지며, 패킷 변조, 도청 및 해킹과 같은 다양한 보안 위협에 노출된다. 따라서 패킷 암호화가 필수이며, 이는 간단한 XOR 부터 AES-128 과 같은 강력한 암호 알고리즘을 활용한다. AES-128 은 강력한 보안을 제공하지만, 멀티플레이어 게임 환경에서는 높은 계산량과 리소스 사용으로 인해 통신 지연을 유발할 수 있다. 따라서, 강력한 보안을 유지하면서도, 빠르고 가볍게 통신 가능한 경량 암호 알고리즘을 적용해보고자 한다.

1.2 연구 목적

본 연구는 유니티 멀티플레이어 환경에서 경량 암호 통신을 구현하고, 이를 통해 게임 내 통신의 보안과 성능 간의 균형을 확보하는 것을 목적으로 한다. 이를 위해 기존의 AES-128 과 다양한 경량 암호 알고리즘을 비교 분석하고, 게임 개발에 최적화된 방안을 제시하고자 한다.

2. 프로젝트 소개 및 설계

- 전체 설계

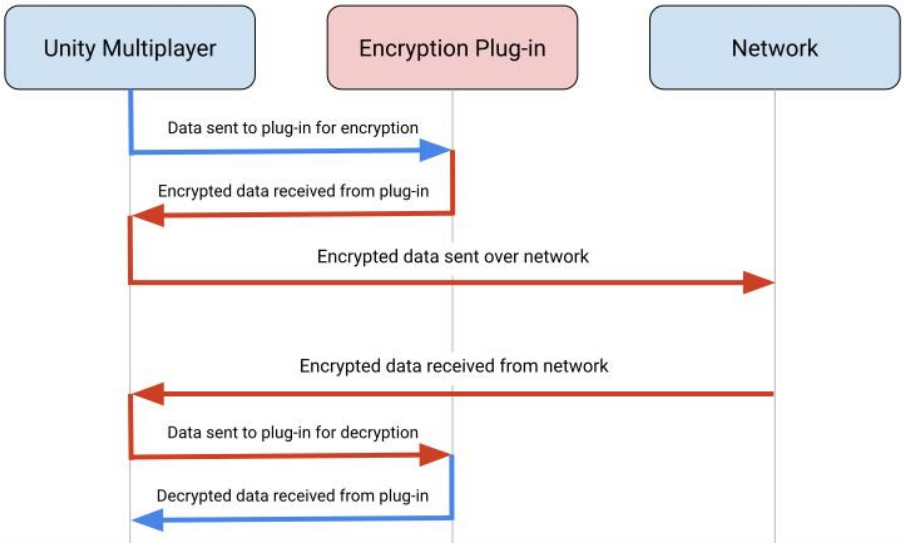


유니티 멀티플레이어 환경은 클라이언트-서버 통신으로 구현한다. 클라이언트는 유니티의 네트워킹 기능을 통해 서버에 접속하고 데이터를 송수신한다. 서버는 클라이언트의 요청을 받아 게임 로직을 처리하고 응답을 전송한다. 해당 과정에서 클라이언트는 암호화 플러그인을 거쳐 패킷을 암호화/복호화 한다.

- Client: 유니티 네트워킹 기능을 통해 메인 서버와 채팅 서버에 접속한다. 통신 과정에서 암호화 플러그인을 거쳐 모든 패킷을 암호화/복호화 한다.
- Main Server: 빠른 응답을 요구하며 통신량이 많고 상대적으로 가벼운 캐릭터의 이동과 같은 패킷을 처리한다.
- Chat Server: 메인 서버보다 응답 속도에 덜 민감하며 상대적으로 무거운 채팅 메시지와 같은 패킷을 처리한다.

통신 패킷을 와이어샤크(Wireshark)와 같은 패킷 캡쳐 프로그램을 통해 확인해보며, AES-128 과 경량 암호 알고리즘들을 적용해가며 패킷 별 전송 속도와 로직 처리에 드는 시간을 측정한다. 궁극적으로, 보안 강도 평가와 게임 내 통신 성능을 테스트한다.

- 암호화 플러그인 설계



유니티 멀티플레이어의 경우 암호화 플러그인을 별도로 제공하지는 않으나, 암호화에 관련된 인터페이스를 제공한다.[1] 기본적으로, 네트워크 통신이 이뤄지기 전 Client 에 부착된 플러그인을 통해 암호화가 이루어진다. 이후, 받은 패킷은 플러그인을 통해 복호화되어 클라이언트에 전달되는 방식이다. 이를 통해 AES-128, ARIA, HIGHT, TWINE 등, 각 암호별 플러그인을 제작 후 성능을 테스트한다.

[1] 멀티플레이어 암호화 플러그인: https://docs.unity3d.com/kr/2023.2/Manual/UNetEncryptionPlugins.html

3. 개발 일정 계획 및 담당 업무

- 개발 일정

5 월	6 월	7 월	8 월	9 월
선행 연구 분석 및 착수 보고서 작성				
	각 암호 알고리즘 구현 및 블록 암호 운영 모드 선택			
		유니티 멀티 플레이어 환경 구축		
			암호화 플러그인 개발 및 적용, 성능 측정	
				최종 보고서 작성 및 발표 준비

- 담당 업무

이름	업무
고세화	유니티에서 사용할 암호 통신 플러그인 제작, ECC 및 TWINE 알고리즘을 조사 후 구현함. 멀티플레이 환경을 구축 후 각 암호 알고리즘 적용 및 암호별 성능 비교를 수행함.
김재민	유니티 3D 엔진을 포함한 게임 엔진조사 및 환경 구축, 이후 HIGHT 알고리즘을 조사 및 패킷 암호화와 관련된 프로그래밍 언어로 구현하고 이를 이용해 유니티 네트워크 환경 구현 보조작업을 수행한다.
이강인	실시간 상호작용이 많은 유니티에서 경량화 암호 알고리즘을 통한 보안 강화 필요성 조사 , AES, ARIA 알고리즘 C# 구현, 이후 경량화 암호 알고리즘간 특징 및 성능비교.