

System Model (Sequence Diagram)
Document

Project Name	인공위성 통신 보안 시뮬레이터
-----------------	------------------

13 조

202002561 조영민

202002546 임우진

202002493 박민서

지도교수: 장진수 교수님

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2023/05/03	문서 작성	임우진

Table of Contents

1.	INTRODUCTION	5
1.1.	OBJECTIVE	5
2.	USE CASE DIAGRAM	6
3.	SEQUENCE DIAGRAM	7
4.	AI 도구 활용 정보	13

List of Figure

그림 1. 유스케이스 다이어그램	6
-------------------------	---

1. Introduction

1.1. Objective

이 명세서는 위성 통신 환경의 핵심 요소를 경량화된 방식으로 재현하고, 다양한 사이버 보안 공격 및 대응 시나리오를 구현하기 위한 '인공위성 통신 보안 시뮬레이터' 개발의 필요성과 방향성을 제시하는 데 그 목적이 있습니다.

현재 위성 통신은 상업 및 국방 분야의 핵심 인프라로 자리잡고 있으나, 기존 시뮬레이터들은 실제 통신환경의 전파지연, 도플러 효과 등 물리적 현상을 제대로 반영하지 못하고, 보안 측면에서도 외부 라이브러리에 의존하는 한계가 있습니다. 이 명세서는 HackRF 기반 RF 통신 환경과 gr-leo 모듈을 활용하여 실제 위성 통신 환경의 물리적 특성을 반영하고, 제밍, 스푸핑, 중간자 공격 등 다양한 보안 공격 시나리오를 구현하여 통합된 보안 시뮬레이션 환경을 제공하는 시스템 개발 계획을 담고 있습니다.

이를 통해 사용자들이 위성 통신 시스템의 보안 취약점을 직접 체험하고 분석할 수 있는 교육 및 연구 도구를 개발하는 것이 궁극적인 목표입니다.

이 문서의 목적을 개략적으로 기술한다. 예를 들어, 이 문서는 시스템 모델 (시퀀스 다이어그램)에 대한 내용을 기술하고 있다. 요구사항 명세 단계에서 작성한 유스케이스 다이어그램을 기반으로 각 유스케이스의 상세한 내부 동작 흐름을 시퀀스 다이어그램으로 모델링한다.

2. Use Case Diagram

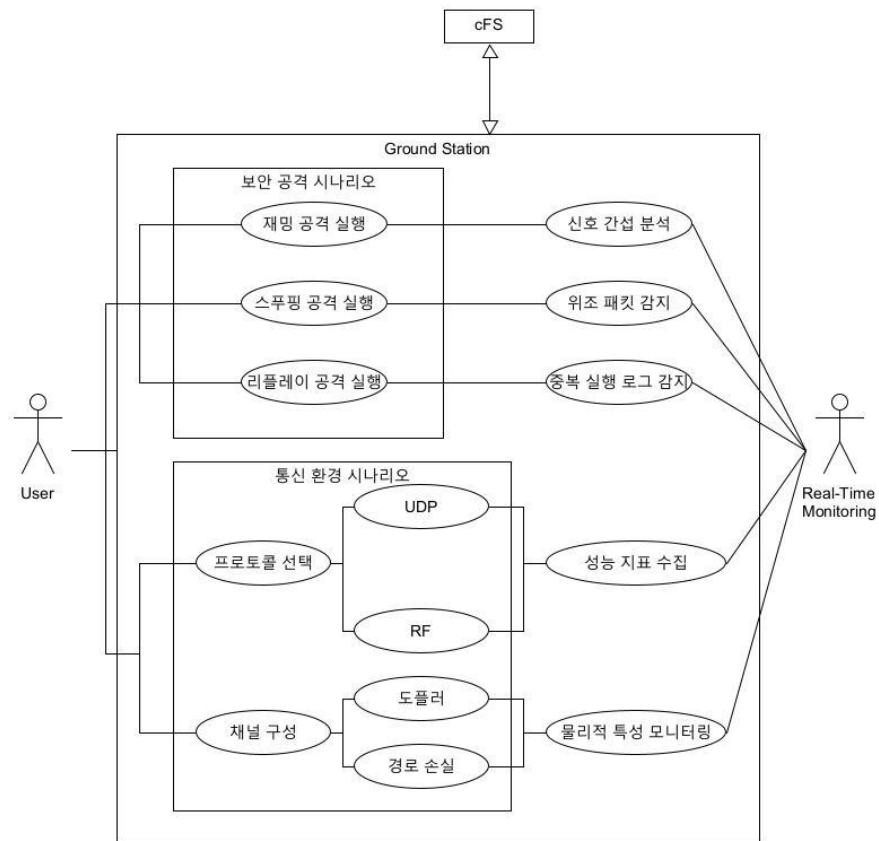
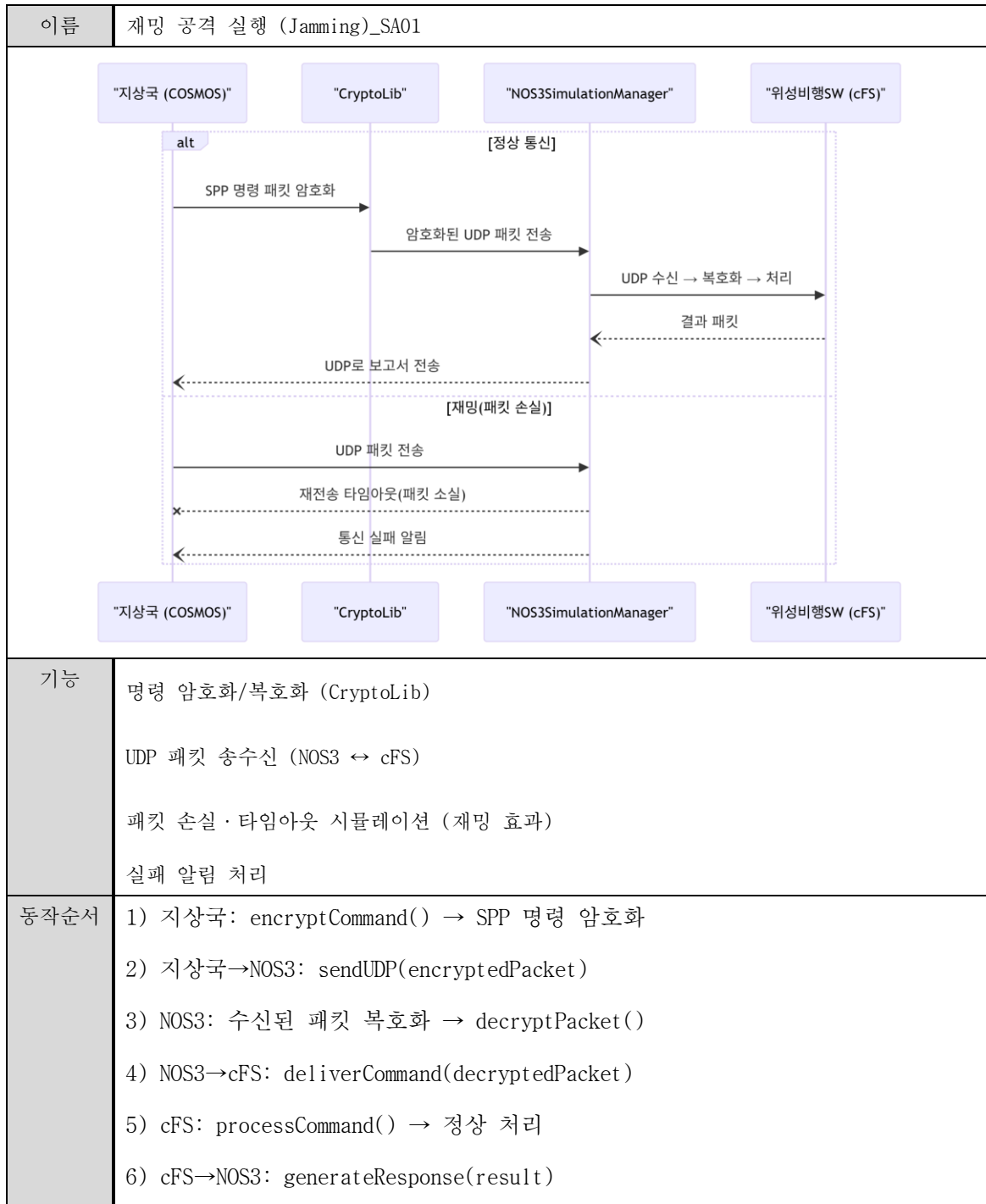


그림 1. 유스케이스 다이어그램

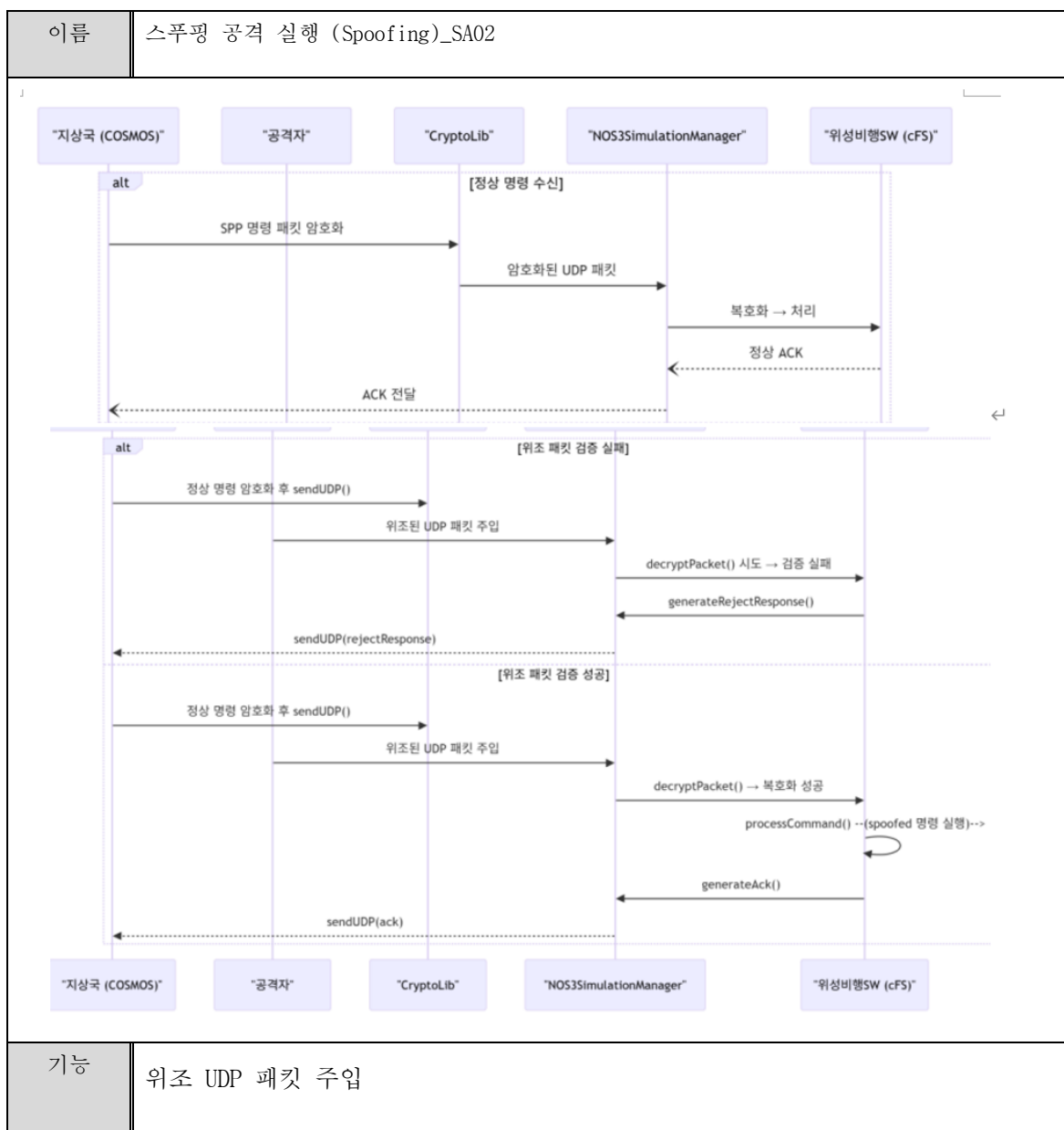
3. Sequence Diagram

3.1 재밍 공격 실행 (Jamming)



	<p>7) NOS3→지상국: sendUDP(responsePacket)</p> <p>8) 재밍(패킷 손실) 시</p> <ul style="list-style-type: none"> - NOS3: 패킷 미수신 감지 → timeout - NOS3→지상국: notifyFailure()
--	--

3.2 스푸핑 공격 실행 (Spoofing)



	<p>복호화 검증 실패 처리</p> <p>거부 응답 전송</p>
동작순서	<p>1) 지상국: 정상 명령 암호화 후 sendUDP()</p> <p>2) 공격자: 위조 명령 암호화 없이 injectUDP(spoofedPacket)</p> <p>3) NOS3: 먼저 도착한 위조 패킷 수신 → decryptPacket() 시도</p> <p>4) cFS: 검증 실패 → rejectCommand()</p> <ul style="list-style-type: none"> - cFS→NOS3: generateRejectResponse() - NOS3→지상국: sendUDP(rejectResponse) <p>5) cFS: 검증 성공 → processCommand() → generateAck() → sendUDP(ack)</p>

3.3 리플레이 공격 실행 (Replay)

이름	리플레이 공격 실행 (Replay)_SA03
<pre> sequenceDiagram participant G as "지상국 (COSMOS)" participant N as "NOS3SimulationManager" participant C as "위성비행SW (cFS)" participant A as "공격자" G->>N: 현행 명령 UDP 전송 N->>C: 복호화 → 처리 → ACK 반환 C-->>N: ACK 전달 N->>A: 패킷 내용 녹취 A->>N: 녹취된 UDP 패킷 재전송 N->>C: 재생 복호화 → (실행 or 거부) C-->>N: (ACK or 거부) N-->>G: 재플레이 결과 보고 </pre>	
기능	정상 패킷 녹취(로그)

	<p>녹취된 UDP 패킷 재전송</p> <p>재생 명령 검증 · 처리</p>
동작순서	<p>1) 지상국: 정상 명령 암호화 → sendUDP()</p> <p>2) NOS3: decryptPacket() → processCommand() → ACK 생성</p> <p>3) NOS3: 정상 패킷 로그(logPacket())</p> <p>4) NOS3→지상국: ACK 전송</p> <p>5) 공격자: replayUDP(loggedPacket)</p> <p>6) NOS3: decryptPacket() → 재생 명령 처리 또는 거부</p> <p>7) NOS3→지상국: 재생 결과 전송</p>

3.4 프로토콜 선택 (Protocol Selection)

이름	프로토콜 선택 (Protocol Selection)_SA04
	<pre> sequenceDiagram participant G as "지상국 (COSMOS)" participant C as "CryptoLib" participant N as "NOS3SimulationManager" participant W as "위성비행SW (cFS)" alt [UDP 선택] G->>C: 테스트 명령 암호화 C->>N: UDP 패킷 전송 N->>W: UDP 수신 → 복호화 → 처리 and [TCP 선택] G->>C: 테스트 명령 암호화 C->>N: TCP 연결 수립 → 데이터 전송 N->>W: TCP 수신 → 복호화 → 처리 and [RF 선택 (미래 확장)] G->>C: 테스트 명령 암호화 C->>N: RF 인터페이스 호출 → 무선 전송 N->>W: RF 수신 → 복호화 → 처리 end W-->>N: 테스트 결과(처리 시간·성공률) N-->>G: 성능 지표 보고 </pre>
기능	<p>다중 프로토콜 지원: UDP, TCP, RF(미래 확장)</p> <p>명령 암호화/복호화: CryptoLib을 이용한 SPP 프레임 암호화 및 복호화</p> <p>연결 관리: TCP 세션 수립·종료, UDP 비연결형 전송, RF 인터페이스 호출</p> <p>성능 측정: 처리 지연(latency), 성공률(reliability) 수집 및 리포트</p>
동작순서	<ol style="list-style-type: none"> 1) 프로토콜 설정: <code>UI → PM</code>에 <code>protocolType</code> 저장 2) 명령 암호화: <code>CryptoLib.encryptCommand(command) → encryptedPacket</code> 3) 전송 <ul style="list-style-type: none"> - UDP: <code>NOS3.sendUDP(encryptedPacket)</code> - TCP: <code>NOS3.openTCP(); sendTCP(encryptedPacket); closeTCP()</code> - RF(확장): <code>NOS3.transmitRF(encryptedPacket)</code> 4) 명령 처리: cFS 앱에서 수신 → 복호화 → <code>processCommand()</code> → ACK 생성

	5) 성과 수집 · 보고: NOS3에서 latency · successRate 계산 → ReportGenerator를 통해 암호화 후 지상국에 전송
--	--

3.5 채널 모델링 (Channel Modeling)

이름	채널 모델링 (Channel Modeling)_SA05
<pre> sequenceDiagram participant User as "사용자" participant NOS3 as "NOS3SimulationManager" participant Ground as "지상국 (COSMOS)" participant Crypto as "CryptoLib" participant Satellite as "위성비행SW (cFS)" User->>NOS3: 채널 파라미터 설정 (지연, 손실률) NOS3->>Ground: 네트워크 시뮬레이션 파이프라인 갱신 activate NOS3 NOS3->>NOS3: deactivate NOS3 Ground->>Crypto: SPP 테스트 명령 암호화 Crypto->>Ground: UDP 패킷 전달 (지연·손실 적용) Ground->>Satellite: 복호화 → 처리 Satellite-->>Ground: 결과 Ground-->>NOS3: Channel-Adjusted 보고 NOS3-->>User: </pre>	
기능	<p>네트워크 지연 및 손실 설정</p> <p>UDP 시뮬레이션 파이프라인 갱신</p> <p>채널 영향 반영된 송수신</p>
동작순서	<ol style="list-style-type: none"> 1) 사용자: setChannelParams(delay, lossRate) 요청 2) NOS3: 네트워크 시뮬레이터에 파라미터 적용 → updateNetworkModel() 3) 지상국: 테스트 명령 암호화 → sendUDP() 4) NOS3: 네트워크 모델 적용 후 지연/손실 시뮬레이션 → simulateNetwork() 5) NOS3: 정상 또는 지연된 패킷 decryptPacket() → processCommand()

	6) cFS: executeCommand() → 결과 반환
	7) NOS3→지상국: 채널 보정 결과 전송

4. AI 도구 활용 정보

사용 도구	<i>GPT-4, Perplexity</i>
사용 목적	<i>NOS3 & cFS 최신 통합 구조 · 인터페이스 확인</i>
프롬프트	<ul style="list-style-type: none"> ● 프로토콜 선택 유스케이스의 동작 순서를 간단히 정리해줘 ● NOS3 cFS 통합 구조 문서 찾아줘
반영 위치	<i>1. 시퀀스 다이어그램 (3.1 - 3.5)</i>
수작업	있음(논리 보강 및 문맥 수정)
수정	