

코드 난독화 시퀀스 다이어그램

System Model (Sequence Diagram) Document

Project Name	코드 난독화 도구 제작
-----------------	--------------

15 조

202002562 조인우

202002508 손지웅

201902686 노형우

지도교수: 조은선 교수님 (서명)

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2025/04/29	초안 작성	조인우
2	2025/05/01	시퀀스 다이어그램 수정	손지웅
3	2025/05/02	오탈자 수정	노형우

Table of Contents

1.	INTRODUCTION	5
1.1.	연구 배경	5
1.2.	연구 목적	5
1.3.	연구 질문/ 가설	5
2.	USE CASE DIAGRAM	7
2.1.	소프트웨어 활용 사례	7
2.2.	문제 해결에 대한 사용 사례 DIAGRAM	7
3.	SEQUENCE DIAGRAM	8
3.1.	해결 방법에 대한 알고리즘 순서도	8
4.	AI 도구 활용 정보	9

List of Figure

그림 1. 소프트웨어 활용 사례 다이어그램	7
그림 2. 문제 해결에 대한 사용 사례 다이어그램	7
그림 3. 시퀀스 다이어그램	8

1. Introduction

1.1. 연구 배경

소프트웨어 산업은 불법 복제와 리버스 엔지니어링 등을 통한 지식 재산권 침해 문제에 지속적으로 직면하고 있으며, 특히 악의적인 공격자가 내부 알고리즘이나 시스템 구조를 분석하는 사례가 증가하고 있다. 최근에는 ChatGPT와 같은 대형 언어 모델(LLM)을 이용한 역난독화 기술까지 등장함에 따라 기존 난독화 기법의 효과가 급격히 저하되고 있으며, 이는 소프트웨어 보안을 위해 보다 정교하고 진화된 난독화 기술의 필요성을 시사한다. 기존 난독화는 특정 패턴에 기반하여 적용되며, 숙련된 분석자나 AI 기반 도구에 의해 쉽게 우회될 수 있어, 단순한 가독성 저하를 넘어 구조적으로 코드 해석을 방해하는 기술적 접근이 요구되고 있다. 이에 본 연구는 컴파일러 기술을 바탕으로 소스 코드 및 중간 표현(IR) 수준에서 역공학을 어렵게 만드는 난독화 전략을 설계하는 것을 목적으로 한다.

1.2. 연구 목적

본 연구의 목적은 코드 난독화 도구 제작을 통해 역공학 및 AI 기반 코드 분석 기술에 대한 방어 수단을 제공하고, 고도화된 코드 보호 효과를 검증하는 것이다. 특히, 보안이 중요한 소프트웨어 개발자 및 기업의 소스 코드 무단 유출 및 불법 복제에 대한 우려를 해소하고, ANTLR 기반 소스 코드 파스트리 분석과 LLVM 기반 중간 표현 수준의 구조적 코드 변환 기법의 효과성을 단계적으로 검증함으로써 실용적이고 기술적으로 의미 있는 난독화 도구 개발에 기여하는 것을 목표로 한다. 이를 통해 대형 언어 모델(LLM)을 포함한 고성능 복원 도구에 대한 저항성을 확보하고, 다양한 언어와 플랫폼에 적용 가능한 범용 난독화 프레임워크 구축이라는 사회적·기술적 의의를 도출하고자 한다.

1.3. 연구 질문/ 가설

본 연구는 다음과 같은 연구 질문에 답하고자 한다:

- RQ1.

제안하는 ANTLR 또는 LLVM 기반 선택적 난독화 도구는 기존 난독화 도구에 비해 AI 기반 복원 도구 (예: LLM, 디컴파일러 등)에 대한 저항성을 얼마나 향상시키는가?

- **RQ2.**
소스 코드 수준 난독화와 IR 수준 난독화 중, 어떤 방식이 실행 성능과 보안성의 균형 측면에서 더 효과적인가?

본 연구는 다음과 같은 가설을 설정할 수 있다:

- **H1.**
제안된 난독화 도구는 기존 난독화 방식에 비해 AI 기반 분석 도구에 대한 복원 저항성을 유의미하게 향상시킬 것이다.
- **H2.**
ANTLR 기반 소스 코드 난독화와 LLVM 기반 IR 난독화는 각각의 적용 대상과 상황에 따라 보안성과 성능 측면에서 상이한 효과를 보일 것이다.

2. Use Case Diagram

2.1. 소프트웨어 활용 사례

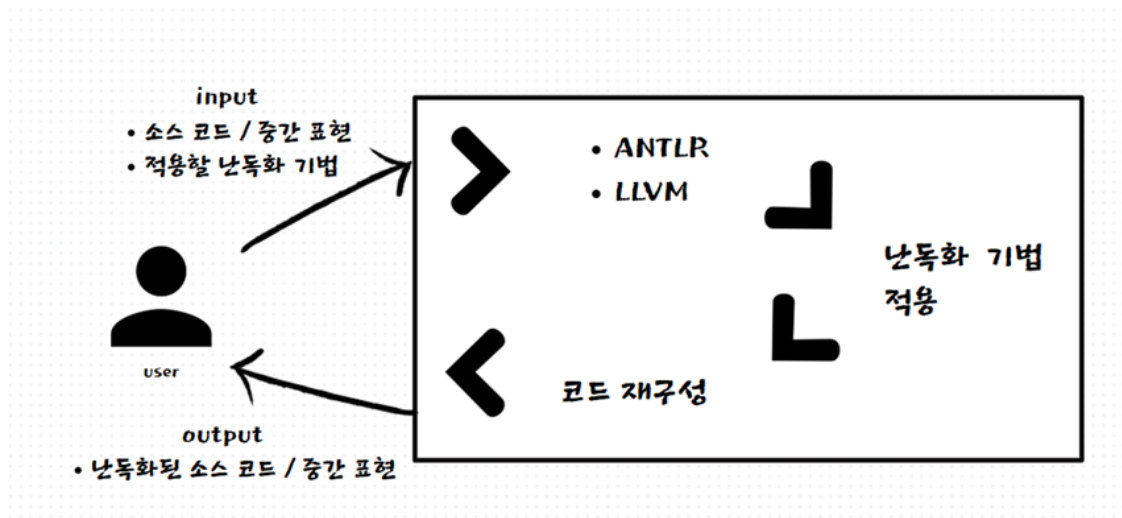


그림 1. 소프트웨어 활용 사례 다이어그램

2.2. 문제 해결에 대한 사용 사례 Diagram

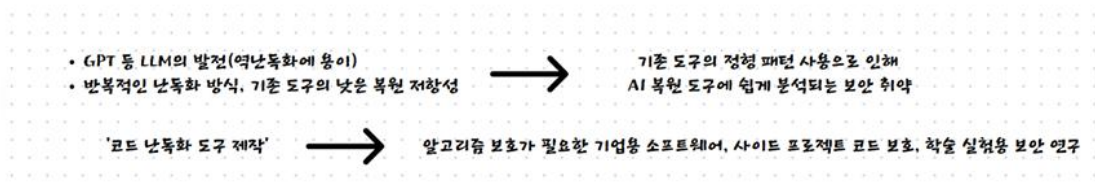
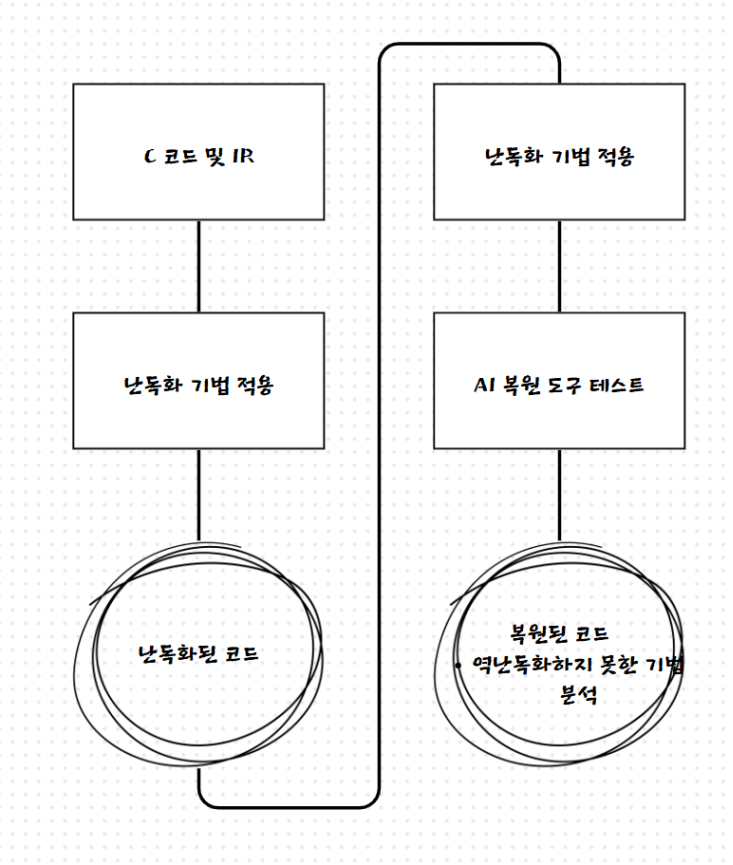


그림 2. 문제 해결에 대한 사용 사례 다이어그램

3. Sequence Diagram

3.1. 해결 방법에 대한 알고리즘 순서도

<p>연구가설 (or 연구질문)</p>	<p>H1. 제안된 난독화 도구는 기존 난독화 방식에 비해 AI 기반 분석 도구에 대한 복원 저항성을 유의미하게 향상시킬 것이다.</p> <p>H2. ANTLR 기반 소스 코드 난독화와 LLVM 기반 IR 난독화는 각각의 적용 대상과 상황에 따라 보안성과 성능 측면에서 상이한 효과를 보일 것이다.</p>
	 <p>그림 3. 시퀀스 다이어그램</p>
<p>핵심 문제 정의</p>	<p>LLM 기반 분석 도구와 고도화된 리버스 엔지니어링에 대응하기 위해, 소스 코드 및 중간 표현(IR) 수준에서 문법적 구조와 실행 흐름을 교란하는 컴파일러 기반 난독화 기법을 설계하고, 이를 통해 구조적 복원 저항성을 확보하는 범용 난독화 프레임워크를 구현한다.</p>

알고리즘 순서	<ol style="list-style-type: none"> 1. 입력 데이터 수집 <ul style="list-style-type: none"> - 사용자 소스 코드 또는 LLVM 중간 표현(IR) 2. 분석 대상 분류 <ul style="list-style-type: none"> - ANTLR 기반 : C 코드 - LLVM 기반 : 중간 표현(IR) 3. 난독화 기법 적용 <ul style="list-style-type: none"> - ANTLR : 식별자 치환, dead code 삽입, 조건 분기 왜곡 등 - LLVM : control flow flattening, opaque predicate 등 4. 코드 재구성 및 출력 <ul style="list-style-type: none"> - 난독화된 결과물 생성 - 원래 코드와 기능 동등성을 유지함 5. AI 복원 도구에 대한 테스트 <ul style="list-style-type: none"> - GPT/LLM 기반 코드 복원 도구에 입력 - 복원을 및 실행 시간 분석
---------	---

4. AI 도구 활용 정보

사용 도구	GPT-4o
사용 목적	알고리즘 순서도를 글로 서술
프롬프트	<ul style="list-style-type: none"> ● 이 그림을 보고 글로 서술해줘.
반영 위치	1. 3.1 해결 방법에 대한 알고리즘 순서도 (p.9)
수작업 수정	있음(의도하지 않은 내용 수정)