P R O J E C T D O C U M E N T

보이지만 읽을 수 없다.

문제정의서(연구개발계획서)

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	2025/04/01	초안 작성	조인우
2	2025/04/02	세부 내용 수정	손지웅
3	2025/04/03	오탈자 수정, 인터뷰	노형우

Table of Contents

List of Figure

1. 연구 개발의 필요성

소프트웨어 산업에서는 불법 복제와 역공학(Reverse Engineering)에 의한 저작권 침해가 지속적으로 문제가 되고 있으며, 이는 기업에 직접적인 경제적 피해를 초래하고 기술 유출의 위험까지 동반한다. 특히 내부 알고리즘과 구조를 파악하는 공격이 증가하고 있으며, 최근에는 LLM와 같은 AI 기반 도구를 활용한 역난독화 기술까지 발전해 기존의 난독화 기법이 무력화되고 있다. 이에 따라 보다 진화된 난독화 기술의 필요성이 대두되고 있으며, 본 연구에서는 컴파일러의 코드 변환 기술을 활용해 기존 소스 코드뿐 아니라 중간 코드 수준에서도 분석을 어렵게 만드는 새로운 난독화 도구를 제안하고자 한다. 이는 게임, 보안 소프트웨어, 기업용 솔루션 등 다양한 분야에서 코드 보안을 강화하고, 역공학 기반 공격에 대한 실질적인 대응 수단을 제공함으로써 기술적·사회적 측면에서 매우 중요한 연구개발 과제이다.

2. 연구 개발의 목표 및 내용

본 프로젝트의 궁극적인 목표는 사람뿐 아니라 최근 빠르게 확산되고 있는 대형 언어 모델(LLM), 예컨대 ChatGPT와 같은 인공지능조차 분석하기 어려운 수준의 난독화 기법을 고안하고, 이를 코드 자동 변환 도구로 구현하는 데 있다. 특히 기존 난독화 기법들이 일정한 패턴을 가지며 AI 기반의 역공학 도구에 의해 쉽게 복원될 수 있는 한계를 넘어서기 위해, 원본 소스 코드를 입력받아 구조적 분석을 통해 난독화된 결과물을 출력하는 자동화 도구 개발을 목표로 한다. 이를 위해 ANTLR의 리스너 기능을 활용해 파스트리(parse tree)를 생성하고, 각 노드를 세밀하게 탐색하여 코드 레벨에서 난독화 처리를 수행한다. 사용자는 이 도구를 통해 코드 보안을 손쉽게 강화할 수 있으며, 불법 복제와 원본 소스의 유출로부터 소프트웨어를 보호하는 실질적인 효과를 기대할 수 있다. 초기에는 C 언어를 대상으로 시작하지만, 향후에는 IR(중간 코드) 수준이나 다양한 언어로의 확장을 통해 보다 범용적이고 강력한 난독화 도구로 발전시킬 계획이다.

이를 달성하기 위한 브레인 스토밍 결과이다.

그림 1. 주요 개념 및 핵심 키워드 도출 과정

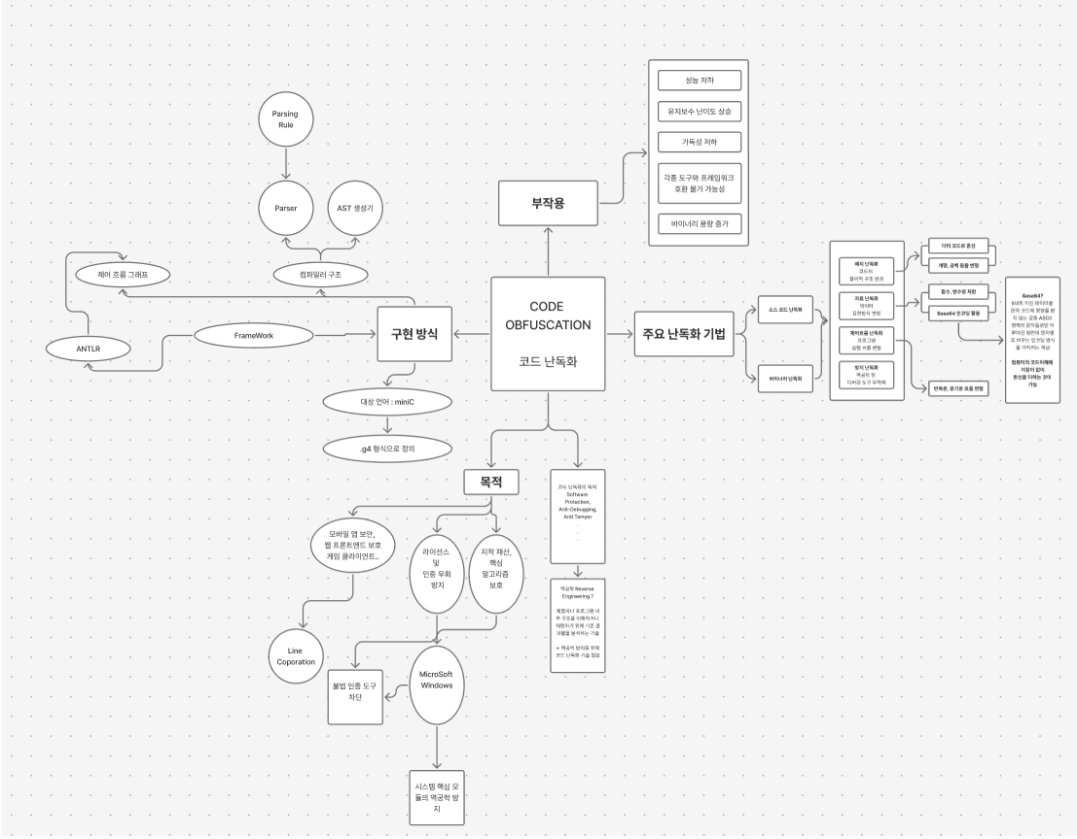

그림 2. 난독화 기법 구성 요소 간의 관계를 정리한 마인드맵

3. 이해당사자 인터뷰/ 설문 인사이트

질문

1. 난독화에 대해 들어본 적이 있나요?
2. 접해보았다면 어디서 어떻게 접하였으며 그 효과는 어느정도였나요?
3. 현재 업계에서 사용되는 난독화에 부족한 점이 있다고 생각해본 적 있나요?
4. 웹 상에서 사용가능한 난독화 도구가 생긴다면 사용해볼 의향이 있나요?
5. 현재 난독화를 사용하지 않는 이유?
6. 만약 새로운 난독화 기법으로 난독화 도구가 제작된다고 하면 사용해 볼 의향이 있나요?

Interviewee A. 28세 중견기업 백앤드 개발자, 4월 4일 인터뷰 진행

1. 난독화에 대해 들어본 적이 있나요?

- 코드를 사람이 읽기 어렵게 변형시키는 기법임을 알고 있다.
- 보통 악의적인 사용자를 방지하기 위해서, 혹은 지적 재산 보호 차원에서 많이 쓰이는 것 같다.
- JavaScript 웹 애플리케이션에서도 클라이언트 코드 노출을 막으려고 난독화를 많이 한다.
- 실제로 회사에서 직접 난독화를 적용한 적은 없지만, 외부 라이브러리나 SDK를 분석할 때 난독화된 코드를 접한 적은 있어서 그 어려움을 체감해 본 경험이 있다.

2. 접해보았다면 어디서 어떻게 접하였으며 그 효과는 어느정도였나요?

- 클라이언트 단에 적용되는 보안 모듈을 외부 보안 업체로부터 받았다. 이때 해당 업체로부터 받은 C 기반 보안 모듈 연동 작업할 때 접해보았다.
- 라이브러리와 연동하면서 문제가 발생했을 때, 디버깅에 어려움을 겪었던 기억이 있다.

- 취약점 분석을 하는 데 소요되는 시간과 노력을 확실히 늘렸다.
- 완벽히 막을 수는 없지만 보안 장벽을 하나 더 두어 보안을 강화한다는 느낌이 들었다.

3. 현재 업계에서 사용되는 난독화에 부족한 점이 있다고 생각해본 적 있나요?

- 정형화된 패턴 → 숙련된 분석자에겐 뚫린다고 생각한다. 특히 난독화의 보안 효과가 시간이 지남에 따라 빠르게 퇴색되는 것은 난독화의 아쉬운 점 중 하나이다.
- 난독화로 인해 불필요하게 복잡해진 제어 흐름이나 과도한 함수 분할/삽입 등이 실제 실행 성능에 영향을 줄 수 있다. 실제로 실행 속도가 10~20% 느려지는 느낌이 들었다.

4. 웹 상에서 사용가능한 난독화 도구가 생긴다면 사용해볼 의향이 있나요?

- 사이드 프로젝트나 프로토타입 단계에서 코드 유출을 방지하려는 목적으로 활용해 볼 수

있을 것 같다.
- 빠른 테스트 용도로 활용해보아도 괜찮을 것 같다.
- 하지만 웹 기반 도구라면 보안이 우선 걱정되며 난독화 품질에 대한 우려 또한 존재한다.

5. 현재 난독화를 사용하지 않는 이유는 무엇인가요?
- 실제 실무에서 난독화는 우선순위가 낮거나 적용이 번거롭고,
난독화로 인해 디버깅이나 유지보수에 부담이 생길 수 있어서 피하게 되는 경우가 많은 것 같다.

6. 만약 새로운 난독화 기법으로 난독화 도구가 제작된다고 하면 사용해 볼 의향이 있나요?
- 기존 방식이 뚫리는 걸 많이 봐서, 기존과 다른 접근 방식이 있다면 사용해 보고 싶다.
- 실행 안정성, 빌드 환경과의 호환성, 그리고 디버깅 가능성이 사용여부 결정에 큰 영향을 미칠 것 같다.

Interviewee B. 53세 비전공자 게임업계 프로그래머, 4월 5일 인터뷰 진행

1. 난독화에 대해 들어본 적이 있나요?
- 비전공자이니만큼 전문적으로 다뤄본 건 아니지만, 업무 중에 보안 관련 회의나 문서에서 종종 언급되는 것을 확인한 적이 있다.

2. 접해보았다면 어디서 어떻게 접하였으며 그 효과는 어느정도였나요?
- 예전에 외부에서 유입된 게임 모듈 중에 난독화가 적용된 Lua 스크립트를 본 적이 있었다.
- 코드를 분석해서 무언가 수정하려는 시도 자체를 포기하게 만드는 느낌이 들었다.

3. 현재 업계에서 사용되는 난독화에 부족한 점이 있다고 생각해본 적 있나요?
- 난독화가 적용되면 읽거나 분석이 어려워지는만큼 디버깅이나 문제 해결이 거의 불가능하다는 점이 단점이라고 생각한다.
- 실제로 당시 팀 회의에서 난독화가 너무 심해서 결국 처음부터 새로 짜는 게 빠르겠다는 이야기가 나온 적이 있었다.

4. 웹 상에서 사용가능한 난독화 도구가 생긴다면 사용해볼 의향이 있나요?
- 요즘은 툴 스크립트를 Python이나 Lua로 간단히 짜는 경우가 많다 보니, 그걸 웹에서 손쉽게 난독화해서 배포할 수 있다면 굉장히 편리할 것 같다.
- 하지만 웹 기반인 경우 역시 보안이 가장 우려된다.
- 또한 난독화 결과물이 정상적으로 돌아가는지 확인할 수 있는 기능이 있어야 마음 놓고

쓸 수 있을 것 같다.

5. 현재 난독화를 사용하지 않는 이유?

- 난독화 자체는 필요하다고 느끼지만, 담당하는 분야에서는 직접 적용할 일이 거의 없다.
- 난독화 적용 시 툴이나 빌드 프로세스를 조정해야 하는데 시간이 걸리고 리스크도 있어서 사용하지 않는 것 같다.

6. 만약 새로운 난독화 기법으로 난독화 도구가 제작된다고 하면 사용해 볼 의향이 있나요?

- 앞서 말한 바와 같이 난독화 자체는 필요하다고 느끼므로 기존보다 더 효과적이고 간편하다면 써볼 의향이 있다.
- 툴 기반으로 자동화된 빌드 환경에서 도입하려면 적용 방식이 단순해야 하고, 문제 발생 시 쉽게 원복할 수 있어야 할 것이다.

공통점

1. 난독화의 정의와 목적에 대해 알고 있으며 접해본 경험이 존재한다.
2. 난독화의 한계와 불편한 점 등 단점을 알고 있지만 보안의 보강에 도움이 된다는 점 또한 인지하고 있다.
3. 웹 기반 도구 제작 시 보안성, 안정성, 코드 작동 검증 기능 등이 필요하다고 보았다.
4. 조건이 맞는다면 새로운 난독화 도구를 사용해 볼 의향이 있다.

차이점

1. 도입을 고려하는 부분

- A: 사이드 프로젝트, 클라이언트 보호, 빠른 테스트 용도
- B: 툴 스크립트 보호 목적

2. 우려하는 부분

- A: 성능 저하, 난독화 품질, 디버깅 가능성
- B: 정상 실행 가능여부, 복구 용이성, 빌드 환경 연동성

인사이트

1. 난독화 도구 개발인 만큼 난독화의 품질이 프로젝트의 성과를 좌우한다. 이에 정형적인 기존 방식에서 벗어난 방식을 적용할 필요가 있다.
2. 기본적인 이야기지만 결과로 도출된 코드의 동작성이 보장되어야 한다.
3. 웹을 통해 도구를 제공 시 보안과 적용 속도 또한 감안하여야 한다.
4. 난독화의 효과 또한 중요하지만 난독화 적용에 드는 비용 대비 효과를 생각할 필요가 있다.

4. 기대 효과 및 향후 확장 가능성

본 연구에서 개발하는 코드 난독화 도구는 소프트웨어 보안을 강화하고, 역공학을 통한 지적 재산권 침해를 방지하는 데 기여할 것으로 기대된다. 첫째, 난독화 기법을 적용함으로써 공격자가 코드의 구조를 분석하는 것을 어렵게 만들어 지적 재산권 보호를 강화할 수 있다. 이를 통해 기업과 개발자는 자사의 소프트웨어를 보다 안전하게 배포할 수 있으며, 불법 복제 및 무단 변조로 인한 경제적 손실을 줄일 수 있다. 둘째, 본 연구는 난독화된 코드의 복원 가능성을 평가하기 위한 기준을 설정하고, 역난독화 도구(예: ChatGPT와 같은 대규모 언어 모델)를 활용하여 난독화된 코드가 원본과 최대한 다르게 표현되도록 설계한다. 이를 통해 역난독화 기술이 발전하더라도 난독화된 코드의 안전성을 유지할 수 있도록 한다. 마지막으로, 본 연구는 ANTLR을 활용하여 파스트리 기반으로 난독화를 수행하는 구조이므로, 향후 C 및 IR뿐만 아니라 컴파일러를 사용하는 다양한 프로그래밍 언어로의 확장이 가능하다. 이러한 확장성을 바탕으로 본 연구는 보다 범용적인 난독화 기법을 개발하는 데 기여할 수 있을 것으로 기대된다.

5. 연구 개발의 추진전략 및 방법

본 연구는 효율적인 코드 난독화 도구 개발을 위해 다음과 같은 단계별 전략과 구체적인 개발 방법론을 바탕으로 진행된다. 먼저, 전체 개발은 당해 학기를 기준으로 네 개의 단계로 나누어 추진된다. 1단계에서는 기존 연구 및 선행 기술에 대한 조사와 관련 기술(컴파일러 이론, 코드 분석, ANTLR 사용법 등)에 대한 사전 학습이 이루어진다. 이어서 2단계에서는 C 언어를 대상으로 하는 난독화 도구의 설계 및 프로토타입 개발이 진행되며, ANTLR을 활용하여 파스트리를 구성하고 각 노드를 기반으로 다양한 난독화 기법을 적용한다. 3단계에서는 C 언어 수준의 소스 코드에서 한 걸음 더 나아가, 바이트 코드 등 중간 표현(IR: Intermediate Representation)을 대상으로 하는 난독화 도구를 설계하고 프로토타입을 개발한다. 마지막으로 4단계에서는 개발된 도구에 대한 테스트와 평가를 수행하며, 특히 ChatGPT와 같은 LLM을 활용하여 역난독화 성능을 테스트하고, 도구의 효과성을 검증한다. 이와 더불어, 프로젝트가 진행되며 도출된 인사이트를 바탕으로 적용 가능한 신규 난독화 기법을 고안하고, 이를 실제 도구에 반영할 계획이다.

기술적으로는 ANTLR을 기반으로 한 파스트리 분석, 구조적 난독화 기법의 적용, 그리고

LLM을 활용한 성능 평가 방식이 병행되어 사용된다. 이를 통해 도구의 자동화 수준과 보안성, 복원 저항성을 정량적으로 측정할 수 있다. 또한 본 연구는 서비스화를 지향하며, 추후 웹 또는 커맨드라인 기반 도구로 배포 가능하도록 설계될 예정이다.

6. AI 도구 활용 정보

사용 도구	GPT-4
사용 목적	인터뷰 질문 초안 작성, 문장 흐름 정리
프롬프트	● 코드 난독화 도구에 대한 인터뷰 진행에 적합한 질문을 나열해줘. ● 이 글을 좀 다듬어줘.
반영 위치	1. 연구 개발의 필요성 2. 연구 개발의 목표 및 내용 3. 이해당사자 인터뷰/설문 인사이트 4. 기대 효과 및 향후 확장 가능성 5. 연구 개발의 추진전약 및 방법
수작업 수정	있음(의도치 않게 추가된 내용 삭제)

7. 참고문헌(Reference)

- KANG, Seoyeon, et al. Obfus: An obfuscation tool for software copyright and vulnerability protection. In: *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. 2021. p. 309-311.
- AONZO, Simone, et al. Obfuscapk: An open-source black-box obfuscation tool for Android apps. *SoftwareX*, 2020, 11: 100403.
- 김효남. 모바일 게임 보안을 위한 게임내 데이터 난독화에 관한 연구. *한국컴퓨터정보학회 학술발표논문집*, 2017, 25.1: 179-180.