

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة العليا للإعلام الآلي - 8 ماي 1945 - سيدي بلعباس
Higher School of Computer Science
8 Mai 1945 - Sidi Bel Abbès



Graduation Thesis

To obtain the diploma of **Engineer's Degree**

Field of Study: **Computer Science**

Specialization: **Computer Systems Engineering (ISI)**

Theme

**Low-Artifact Code Obfuscation via Composable Anti-
Decompilation Primitives**

Presented by
Touahri Khalid

Defended on: **June 23rd, 2026**
In front of the jury composed of

Dr. Malki Abdelhamid
Pr. Benslimane Sidi Mohamed
Dr. Aonzo Simone
Dr. Anani Djihed

President of the Jury
Supervisor
Co-Supervisor
Examiner

Academic Year: 2025/2026

Modern decompilers have significantly lowered the barrier to reverse engineering, producing readable pseudo-code that enables rapid analysis of compiled binaries. While numerous obfuscation techniques exist to impede comprehension, they typically introduce distinctive patterns that make protected code trivially detectable, undermining their practical utility.

This thesis presents four composable anti-decompilation primitives for the x86 architecture that exploit implicit assumptions in decompiler algorithms to hide code from analysis tools while minimizing observable artifacts. The primitives include: state corruption through callee-saved register manipulation to induce dead branch elimination; x86-64 mode switching to exploit instruction encoding differences between 32-bit and 64-bit modes; overlapping instruction encoding to conceal function calls within legitimate instruction operands; and function boundary confusion to prevent function discovery in disassembler listings.

Unlike traditional potency-focused obfuscation that makes code difficult to understand, these primitives achieve concealment-focused protection that renders target code invisible to decompilers. Each primitive has a well-defined artifact profile enabling strategic composition, where multiple primitives can be layered to achieve complete artifact elimination.

A source-to-source transformation tool implementing these primitives is presented, allowing selective application through simple code annotations. Evaluation across IDA Pro, Ghidra, and Binary Ninja demonstrates that all primitives successfully hide target code while maintaining high concealment across local, global, and strategic dimensions.

Les décompilateurs modernes ont considérablement réduit la barrière à la rétro-ingénierie, produisant un pseudo-code lisible qui permet une analyse rapide des binaires compilés. Bien que de nombreuses techniques d'obfuscation existent pour entraver la compréhension, elles introduisent généralement des motifs distinctifs qui rendent le code protégé trivialement détectable, compromettant ainsi leur utilité pratique.

Cette thèse présente quatre éléments de base anti-décompilation composables pour l'architecture x86 qui exploitent les hypothèses implicites des algorithmes de décompilateurs pour cacher le code aux outils d'analyse tout en minimisant les artefacts observables. Ces éléments comprennent : la corruption d'état par manipulation des registres sauvegardés par l'appelé pour induire l'élimination de branches mortes ; le changement de mode x86-64 pour exploiter les différences d'encodage des instructions entre les modes 32 bits et 64 bits ; l'encodage d'instructions chevauchantes pour dissimuler les appels de fonctions dans les opérandes d'instructions légitimes ; et la confusion des limites de fonctions pour empêcher la découverte de fonctions dans les listings du désassembleur.

Contrairement à l'obfuscation traditionnelle axée sur la puissance qui rend le code difficile à comprendre, ces éléments de base réalisent une protection axée sur la dissimulation qui rend le code cible invisible aux décompilateurs. Chaque élément possède un profil d'artefacts bien défini permettant une composition stratégique, où plusieurs éléments peuvent être superposés pour atteindre une élimination complète des artefacts.

Un outil de transformation source-à-source implémentant ces éléments est présenté, permettant une application sélective via de simples annotations de code. L'évaluation sur IDA Pro, Ghidra et Binary Ninja démontre que tous les éléments cachent avec succès le code cible tout en maintenant une haute dissimulation à travers les dimensions locale, globale et stratégique.

لقد سهّلت برامج فك التشفير الحديثة عملية الهندسة العكسية بشكل كبير، إذ تُنتج شفرة زائفة قابلة للقراءة تُمكن من التحليل السريع للملفات الثنائية المجمّعة. ورغم وجود العديد من تقنيات التمويه التي تُعيق الفهم، إلا أنها عادةً ما تُدخل أنماطاً مميزة تجعل من السهل جداً اكتشاف الشفرة المحمية، مما يُقلل من جدواها العملية.

تُقدّم هذه الأطروحة أربع خوارزميات أساسية قابلة للتركيب لمكافحة فك التشفير في معمارية x86، تستغل الافتراضات الضمنية في خوارزميات فك التشفير لإخفاء الشفرة عن أدوات التحليل مع تقليل الآثار المرئية. تشمل هذه الخوارزميات: تشويه الحالة من خلال التلاعب بالسجلات المحفوظة للدالة المُستدعاة لحذف الفروع الميتة؛ والتبديل بين وضعي x86 و64 لاستغلال اختلافات ترميز التعليمات بين الوضعين 32 بت و64 بت؛ وتداخل ترميز التعليمات لإخفاء استدعاءات الدوال داخل معاملات التعليمات المشروعة؛ والتشويش على حدود الدوال لمنع اكتشافها في قوائم مُفكّك الشفرة.

على عكس تقنيات التشفير التقليدية التي تركز على قوة التشفير وتجعل فهم الكود صعباً، توفر هذه التقنيات حماية تركز على إخفاء الكود المستهدف، مما يجعله غير مرئي لبرامج فك التشفير. لكل تقنية ملف تعريف محدد جيداً يُمكن من تركيبها استراتيجياً، حيث يمكن دمج عدة تقنيات لتحقيق إزالة كاملة للكود.

يتم تقديم أداة لتحويل الكود المصدري باستخدام هذه التقنيات، مما يسمح بتطبيقها بشكل انتقائي من خلال إضافة تعليقات توضيحية بسيطة للكود. يُظهر التقييم باستخدام IDA Pro وGhidra وBinary Ninja أن جميع التقنيات تُخفي الكود المستهدف بنجاح مع الحفاظ على مستوى عالٍ من الإخفاء على المستويات المحلية والعالمية والاستراتيجية.