

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 Master's Degree

Field of Study: **Computer Science**

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

Theme

Stealthy Software Obfuscation: A Systematic Review

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

Software obfuscation protects programs against reverse engineering and Man-At-The-End (MATE) attacks by transforming code to impede comprehension. However, traditional obfuscation techniques often introduce detectable artifacts, enabling adversaries to identify protected code and apply targeted deobfuscation. The emergence of Large Language Models (LLMs) in reverse engineering workflows further challenges complexity-focused approaches, as AI-assisted analysis can process obfuscated patterns that would slow human analysts.

We examine stealthy software obfuscation techniques that prioritize concealment alongside potency. We evaluate approaches using the classical four-dimensional criteria (potency, resilience, cost, and stealth) alongside a three-dimensional concealment framework distinguishing local (locator resistance), global (detector resistance), and strategic (learnability resistance) stealth properties. Our coverage spans anti-disassembly methods that exploit tool assumptions, anti-decompilation techniques targeting semantic analysis, and high-concealment approaches including application-specific obfuscation.

We identify a fundamental trade-off between potency and concealment: techniques maximizing comprehension difficulty typically introduce distinctive patterns, while stealthy approaches must constrain transformations to blend with natural code. Methods targeting the binary analysis frameworks (anti-analysis methods) offer the most promising path toward achieving both protection strength and undetectability. We conclude that effective software protection requires defense in depth, combining high-potency transformations for critical regions with stealthy techniques that avoid triggering targeted analysis.

L’obfuscation logicielle protège les programmes contre la rétro-ingénierie et les attaques de type MATE (Man-At-The-End) en transformant le code pour entraver sa compréhension. Cependant, les techniques d’obfuscation traditionnelles introduisent souvent des artefacts détectables, permettant aux adversaires d’identifier le code protégé et d’appliquer une désobfuscation ciblée. L’émergence des grands modèles de langage (LLMs) dans les processus de rétro-ingénierie met davantage à l’épreuve les approches axées sur la complexité, car l’analyse assistée par intelligence artificielle peut traiter des motifs obfusqués qui ralentiraient les analystes humains.

Nous examinons les techniques d’obfuscation logicielle furtive qui privilégient la dissimulation en plus de la puissance. Nous évaluons les approches selon les quatre critères classiques (puissance, résilience, coût et furtivité) ainsi qu’un cadre de dissimulation tridimensionnel distinguant les propriétés de furtivité locale (résistance au localisateur), globale (résistance au détecteur) et stratégique (résistance à l’apprentissage). Notre étude couvre les méthodes anti-désassemblage qui exploitent les hypothèses des outils, les techniques anti-décompilation ciblant l’analyse sémantique, et les approches à haute dissimulation incluant l’obfuscation spécifique aux applications.

Nous identifions un compromis fondamental entre puissance et dissimulation : les techniques maximisant la difficulté de compréhension introduisent généralement des motifs distinctifs, tandis que les approches furtives doivent contraindre les transformations pour se fondre dans le code naturel. Les méthodes ciblant les cadres d’analyse binaire (méthodes anti-analyse) offrent la voie la plus prometteuse pour atteindre à la fois force de protection et indétectabilité. Nous concluons qu’une protection logicielle efficace nécessite une défense en profondeur, combinant des transformations à haute puissance pour les régions critiques avec des techniques furtives qui évitent de déclencher une analyse ciblée.

تحمي تقنيات التمويه البرمجي البرامج من هجمات الهندسة العكسية وهجمات الوصول إلى نهاية البرنامج (MATE) عن طريق تغيير الشيفرة البرمجية بطريقة تعيق فهمها. مع ذلك، غالباً ما تُدخل تقنيات التمويه التقليدية آثاراً قابلة للكشف، مما يُمكن المهاجمين من تحديد الشيفرة المحمية وتطبيق تقنيات إزالة التمويه المُستهدفة. يُشكّل ظهور نماذج اللغة الكبيرة (LLMs) في عمليات الهندسة العكسية تحدياً إضافياً للأساليب التي تُركّز على التعقيد، حيث يُمكن للتحليل المدعوم بالذكاء الاصطناعي معالجة الأنماط المُموّهة التي من شأنها إبطاء عمل المحللين البشريين.

ندرس تقنيات التمويه البرمجي الخفية التي تُعطي الأولوية للإخفاء إلى جانب الفعالية. نُقيّم الأساليب باستخدام المعايير الكلاسيكية رباعية الأبعاد (الفعالية، والمرونة، والتكلفة، والتخفي) إلى جانب إطار عمل ثلاثي الأبعاد للإخفاء يُميّز بين خصائص التخفي المحلية (مقاومة المحدّد)، والعالمية (مقاومة الكاشف)، والاستراتيجية (مقاومة قابلية التعلّم). يشمل نطاق تغطيتنا أساليب مكافحة التفكيك التي تستغل افتراضات الأدوات، وتقنيات مكافحة فك التجميع التي تستهدف التحليل الدلالي، وأساليب الإخفاء العالية بما في ذلك التمويه الخاص بالتطبيقات.

نُحدد مفاضلة جوهرية بين الفعالية والتخفي: فالتقنيات التي تُعظّم صعوبة الفهم تُدخل عادةً أنماطاً مميزة، بينما يجب على الأساليب الخفية تقييد التحويلات لتندمج مع الشيفرة الأصلية. تُقدم الأساليب التي تستهدف أطر تحليل الملفات الثنائية (أساليب مكافحة التحليل) المسار الأكثر جدوى لتحقيق كلٍ من قوة الحماية وعدم إمكانية الكشف. نستنتج أن الحماية الفعالة للبرمجيات تتطلب دفاعاً متعدد الطبقات، يجمع بين تحويلات عالية الفعالية للمناطق الحساسة وتقنيات خفية تتجنب إثارة التحليل المُستهدف.