玉田 春昭（たまだ はるあき）

情報理工学部 准教授
博士（工学）／
奈良先端科学技術大学院大学

ホームページ URL
https://tamadalab.github.io/

主な研究業績

■ 1. 中村 潤，玉田 春昭，“大量のソフトウェアを対象にしたソフトウェアバースマークによる盗用検出 —全文検索システムを用いた検査対象の絞り込み手法”，情報処理学会論文誌，Vol. 61, No. 2, pp. 454--473, February 2020.

■ 2. 大槻 成輝，玉田 春昭，神崎 雄一郎，“JVM 環境におけるオペコード列と名前に着目した適用難読化ツールの特定”，2020 年暗号と情報セキュリティシンポジウム予稿集（SCIS 2020），pp. 1E1-3, January 2020.

■ 3. 玉田 春昭，神崎 雄一郎，“Java バイトコードを対象とした命令の頻度解析による適用難読化ツールの特定”，コンピュータセキュリティシンポジウム 2019 予稿集（CSS 2019），pp. 1C1-3, October 2019.

■ 4. 横井 昂典, 玉田 春昭, “単体テストコードとアスペクト指向を用いた動的バースマークの抽出コストの削減”，情報処理学会論文誌，Vol.60, No.7，July 2019.

■ 5. 磯部 陽介，玉田 春昭，“ランダムフォレストを用いた名前難読化の耐タンパ化性能の評価”，情報処理学会論文誌，Vol.60, No.4, pp. 1063-1074, April 2019.

■ 6. 玉田 春昭，神崎 雄一郎，“オペコードの編集距離を用いた JVM 向け難読化ツールの難読化性能の評価”，2019 年暗号と情報セキュリティシンポジウム予稿集（SCIS 2019），3D2-1，January 2017.

■ 7. Takanori Yokoi, and Haruaki Tamada, “A Beforehand Extraction Method for Dynamic Software Birthmarks using Unit Test Codes,” In Proc. 19th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2018), July 2018.

■ 8. Yosuke Isobe, and Haruaki Tamada, “Are Identifier Renaming Methods Secure? -An Evaluation Focuses on Opcodes using Random Forest-,” In Proc. 19th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2018), July 2018.

■ 9. 西 陽太，神崎 雄一郎，門田 暁人，玉田 春昭，“難読化された Java バイトコードに対するシンボリック実行攻撃の困難さ評価の検討”，第 25 回ソフトウェア工学の基礎ワークショップ（FOSE2018），pp. 151-152，November 2018.

■ 10. Jun Nakamura, and Haruaki Tamada, “mituba: Scaling up Software Theft Detection with the Search Engine”, Proc. International Conference on Software Engineering and Information Management (ICSIM 2018), pp. 6--10, January 2018.

キーワード　プログラム解析、リポジトリマイニング、バイナリ解析

研究テーマ Research theme

バースマークによるプログラム解析技術

概　要 Overview

ソフトウェアの盗用を検出する技術としてソフトウェアバースマーク技術が提案されています。この技術は、バイナリからソフトウェアの特徴を抽出し、ソフトウェア間の類似度を計測します。また、プログラム中の異なる特徴に着目することで、異なるバースマークが形成できます。このバースマークは電子透かしと異なり、事前に情報を埋め込む必要はないため、どのようなソフトウェアに対しても適用可能であり、削除や変更が困難であるような、実行に不可欠な部分に着目します。加えて、発見されていない盗用を検出する目的であることから、大量のソフトウェアの中から特定のソフトウェアと似たソフトウェアを検出するために設計されているため、高速な検査が可能です。

近年、このバースマークは、盗用を検出する目的以外にも、バイナリを解析する技術として用いられています。例えば、名前難読化の堅牢性評価やソフトウェア部品の機能調査などです。名前難読化の堅牢性評価は、名前難読化されたソフトウェアがどの程度保護されているかを確認するため、メソッドの命令列を元にしたバースマークを教師データとして、機械学習により逆変換を試みました（研究業績 2, 3, 5, 8）。その結果、約 40% の動詞が復元できることが確認でき、名前難読化の堅牢性はそれほど高くないことが示されました。

また、動的解析はプログラムを実行させるため、実施が難しいという側面がありました。従来の動的バースマークの抽出も、プログラムに入力を与え実行させる必要があるため、多大なコストが必要でした。しかし、横井らの研究により、対象は限定されるものの、単体テストコードを用いることで、動的バースマークの抽出を自動化できることができました（研究業績 4, 7）。つまり、プログラムの動的解析の自動化がある程度実現できると期待できることを意味します。現在はこの方法を拡張し、単体テストコードを自動生成することで、単体テストが与えられないプログラムに対しても、動的解析の自動化を試みています（研究業績 9）。

一般に、バースマーク分析は高速に処理できますが、検査対象のファイルが数百万、数千万規模になると、やはり検査に多くの時間が必要になります。これを解決するため、全文検索システムを利用したバースマークシステムを構築しました。このシステムは明らかに関係のないファイルを除外してから、従来システムで検査します。その結果、検査時間を 80% 以上短縮でき、検査対象の規模をより増加させることが可能になったと言えます（研究業績 1, 10）。

このように、ソフトウェアバースマークを利用したバイナリ解析を中心とした研究を行なっています。一方で、バースマークを用いつつも、盗用検出ではない解析も実施しています。例えば、難読化ツールは今日いくつかリリースされていますが、難読化の性能は比較されていません。そこに切り込み、逆変換の容易性の第一歩として、どの難読化ツールで難読化されたのかを特定しようと試みています（研究業績 2, 3, 6）。

応用分野 Application areas

・類似モジュールの検出
・ライセンスコンフリクトの検出